#NPS52-90-008

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A223 444

DTIC
ELECTE
JUL 06 1990
S B D

THESIS

SPECIFICATION OF MIL-STANDARD 1553 BUS
PROTOCOL AND APPLICATION TO EA-6B
COMMUNICATIONS COUNTERMEASURES

by

Peter Hugh Christensen

December 1989

Thesis Advisor: Gilbert M. Lundy

90 07 1 5 069

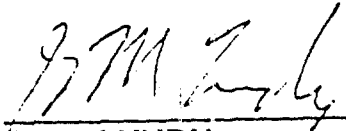## NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R. W. West, Jr.
Superintendent

Harrison Shull
Provost

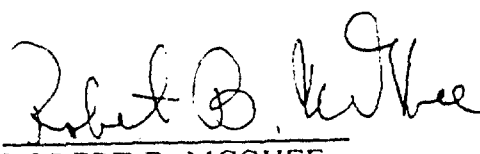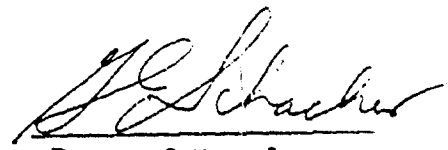Reproduction of all or part of this report is authorized.

G. M. LUNDY
Assistant Professor
of Computer Science

Reviewed by:

Released by:

ROBERT B. MCGHEE
Chairman
Department of Computer Science

Dean of Faculty
and Graduate Studies

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	Approved for public release; distribution is unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S) NPS52-90-008	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 37	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School

6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000

8a NAME OF FUNDING/SPONSORING ORGANIZATION Naval Postgraduate School	8b OFFICE SYMBOL (If applicable) Code 52Ln	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER O&MN, Direct Funding

8c ADDRESS (City, State, and ZIP Code) Monterey, California 93943	10 SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
SPECIFICATION OF MIL-STANDARD 1553 BUS PROTOCOL AND APPLICATION TO EA-6B COMMUNICATIONS COUNTERMEASURES

12 PERSONAL AUTHOR(S)
Christensen, Peter H.

13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM_____ TO_____	14 DATE OF REPORT (Year, Month, Day) 1989, December	15 PAGE COUNT 94

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17	COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Mil-Standard 1553; Specification of Protocols; EA-6B, Theses, Radio Communications, Transmission Lines, Electrical and electronic equipment, Countermeasures

19 ABSTRACT (Continue on reverse if necessary and identify by block number)
A model for the specification and analysis of communication protocols called Systems of Communicating Machines is used to describe and analyze a simplified version of the Mil-Standard 1553 bus protocol. The protocol is used in the EA-6B aircraft for digital communication between aircraft subsystems. The model uses a combination of finite state machines, variables and predicate action tables in the specification of the Bus Controller and Remote Terminals. The enabling predicates determine when a transition may be taken on the finite state machine and actions alter variable values as transitions occur. Normal, error-free 1553 bus command/response information transfers are modeled. The 1553 Mil-Standard does not contain an equivalent specification using a formally defined model.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Professor Gilbert M. Lundy	22b TELEPHONE (Include Area Code) (408) 646-2094	22c OFFICE SYMBOL Code 52Ln

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

#19 - ABSTRACT - (CONTINUED)

Practical application to the EA-6B Prowler is focused upon the requirements for transparent integration of the AN/ASQ-191 Radio Countermeasures Set into the existing aircraft bus architecture. Transparent integration into the tactical jamming system of the aircraft would make the ASQ-191 receiver and jamming operations an integral part of aircraft operation and employment, and permit integrated pre-mission planning with TEAMS.

Specification of Mil-Standard 1553 Bus Protocol
and Application to EA-6B
Communications Countermeasures

by

Peter H. Christensen
Lieutenant Commander, United States Navy
B.A., Linfield College, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1989

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Author: _____
Peter H. Christensen

Approved by: _____
Gilbert M. Lundy, Thesis Advisor

_____
James R. Powell, Second Reader

_____
Robert Mcghee, Chairman
Department of Computer Science

iii
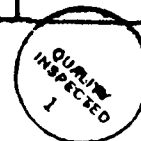
ABSTRACT

A model for the specification and analysis of communication protocols called **Systems of Communicating Machines** is used to describe and analyze a simplified version of the Mil-Standard 1553 bus protocol. The protocol is used in the EA-6B aircraft for digital communication between aircraft subsystems. The model uses a combination of finite state machines, variables and predicate action tables in the specification of the Bus Controller and Remote Terminals. The enabling predicates determine when a transition may be taken on the finite state machine and actions alter variable values as transitions occur. Normal, error-free 1553 bus command/response information transfers are modeled. The 1553 Mil-Standard does not contain an equivalent specification using a formally-defined model.

Practical application to the EA-6B Prowler is focused upon the requirements for **transparent integration** of the AN/ASQ-191 Radio Countermeasures Set into the existing aircraft bus architecture. Transparent integration into the tactical jamming system of the aircraft would make the ASQ-191 receiver and jamming operations an integral part of aircraft operation and employment, and permit integrated pre-mission planning with TEAMS.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I. **INTRODUCTION**

All written languages have characters, words made up of characters and grammars describing valid sequences of words. Characters, words and grammars together formally describe the characteristics of a language. A communications protocol may be described as a language which computers use to exchange data. Communications protocols have their own characteristics which must also be formally defined.

The description of a protocol must be precise, unambiguous and error free. The binary one and zero correspond to the characters of the language. Words are made up of bits. Each sequence of words must follow some predetermined grammatical sequence. As the description of the protocol builds in complexity, it becomes difficult for humans to understand the full meaning of a communications protocol. Therefore several formal models (languages) have been developed to describe communications protocols. Models may depend upon Finite State Machines, Petri Nets or specialized programming languages. Some models are hybrids which borrow from several formal models.

Systems of Communicating Machines (SYSCOM) is a formal model which may be used to describe a communications protocol. The model permits the complex sequence of digital words passed between computers across a common medium to be abstracted into

1

a more understandable form. The communications medium may be represented as a group of shared variables which machines use to exchange data. The meaning of the data stored in the shared variables is determined by a predicate action table. Finally Finite State Machine diagrams are used to model the grammar of the protocol. The added benefit of the formal model is that an analysis of the protocol may be conducted to determine the correctness of the grammar and identify potential problem areas. Chapter I describes the model in detail.

The Mil-Standard 1553 is a protocol used widely among Military Aircraft. [Ref. 1] is published by the Department of Defense as a complete description of the protocol and physical characteristics of the hardware. The background behind the Mil-Standard is briefly described in [Ref. 2:pp. 4-9--4-12]. The 1553 Mil-Standard was born out of necessity. During the 1950's, aircraft weapons systems became too complex to be supported by independent subsystems. During the 1960's, avionics integration created a dramatic increase in the complexity of aircraft subsystems. To permit avionics communication, multiple aircraft subsystems were interconnected with dissimilar I/O ports and complex wiring. The Mil-Standard was first issued in 1973 to address the issue of increasing aircraft complexity. The most current version 1553B, was issued in 1978. Two changes have been submitted subsequently. The 1553 bus architecture permits all

subsystems to communicate through similar I/O ports, across a common bus, utilizing a common protocol. Figure 1 represents the 1553 Bus configuration of the EA-6B ICAP II aircraft.

Although complete, the Mil-Standard is very detailed and in some cases difficult to fully assimilate. For example, Figure 2 shows the bit by bit representation of the format for Command, Status and Data words. There are 17 distinct fields each of which should be understood before attempting to program a system. Table 1, derived from [Ref. 1] describes 16 different types of special mode commands which may be sent to 1553 bus participant. Because the descriptions are written in English, some ambiguity is introduced and misunderstanding may occur.

Formal modeling of the 1553, or any other protocol, provides three important benefits:

1. The formal model can make the functioning of the protocol more abstract and simple to understand.

2. Precise and unambiguous definition of the protocol permits conversion of the model into software, firmware or hardware, with less chance of errors.

3. Analysis of the model ensures that the model functions correctly and that no deadlocks exist.

The **Systems of Communicating Machines Model** may be used to formally specify the 1553 standard, with sufficient detail to promote a full understanding and permit an analysis for correctness. A simplified version of the 1553 protocol expressed in terms of the model is presented in Chapter III and an analysis of the protocol in Chapter IV.

3
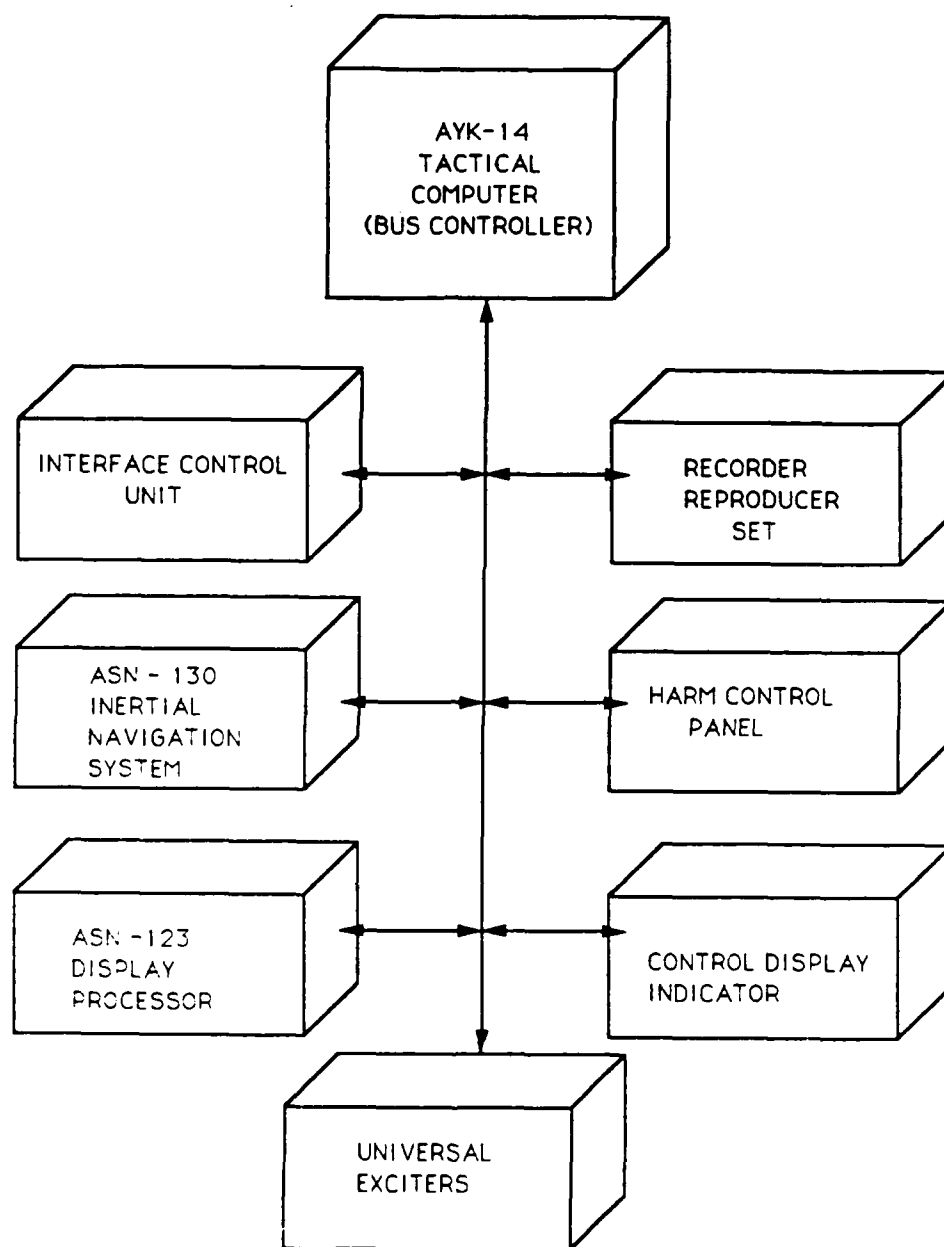
Figure 1.   EA-6B 1553 Data Bus Participants

4

BIT TIMES

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

| SYNC | RT ADDRESS | TR | SUBADDRESS/ MODE | DATA WORD CNT/MODE CD | P |

| SYNC | DATA | P |

| SYNC | RT ADDRESS | | | | | RSVD | | | | | | P |

MESSAGE ERROR
INSTRUMENTATION
SERVICE REQUEST
BROADCAST COMMAND RECEIVED
BUSY
SUBSYSTEM FLAG
DYN. BUS CONTROL ACCEPT.
TERMINAL FLAG
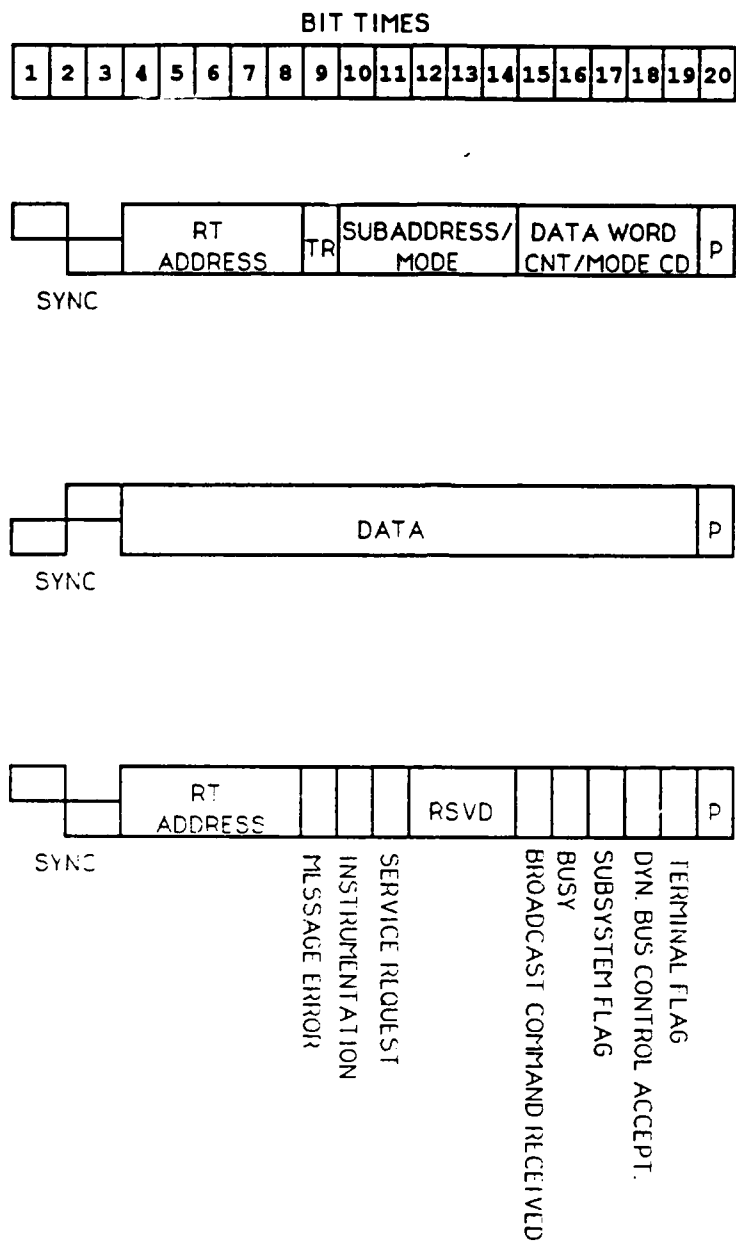
Figure 2. 1553 Word Formats

5

TABLE 1

ASSIGNED MODE CODES

T/R BIT	MODE CODE	FUNCTION	DATA WORD	BROADCAST
1	00000	Dynamic Bus Control	No	No
1	00001	Synchronize	No	Yes
1	00010	Transmit Status Wd	No	No
1	00011	Initiate Self Test	No	Yes
1	00100	Xmitter Shutdwn	No	Yes
1	00101	Ovrd Xmitter Shutdwn	No	Yes
1	00110	Inhibit Term Flag	No	Yes
1	00111	Ovrd Inhibit Term Fl	No	Yes
1	01000	Reset Remote Term	No	Yes
1	01001	Reserved	No	TBD
	to			
1	01111	Reserved	No	TBD
1	10000	Transmit Vector Word	Yes	Yes
0	10001	Synchronize	Yes	Yes
1	10010	Transmit Last Cmd	Yes	No
1	10011	Transmit Bit Word	Yes	No
0	10100	Xmitter Shutdwn	Yes	Yes
0	10101	Ovrd Xmitter Shutdwn	Yes	Yes
1 or 0	10110	Reserved	Yes	TBD
	to			
1 or 0	11111	Reserved	Yes	TBD

The SYSCOM Model of the 1553 may be enhanced with sufficient detail to model timing, errors and special features of the protocol. The resultant predicate/action table is precise enough to be converted into a software program or converted into hardware. Chapter V builds upon the basic model to show how this enhancement may be conducted.

II. SYSTEMS OF COMMUNICATING MACHINES

A. DEFINITION OF THE MODEL

The model which will be used to describe the 1553 Bus protocol is called **Systems of Communicating Machines** (SYSCOM). In this section a brief but formal definition of SYSCOM is presented. A more detailed description is available in [Ref. 3].

A **system of communicating machines** is an ordered pair $C = (M,V)$, where

$$M = \{m_1, m_2, \ldots, m_n\}$$

is a finite set of **machines**, and

$$V = \{v_1, v_2, \ldots, v_n\}$$

is a finite set of **shared variables**. The set V has two designated subsets R_i and W_i which are specified for each machine m_i. The subset R_i of V is called the **read access variables** for machine m_i. The subset W_i is called the set of **write access variables** for m_i.

Each machine m_i contained in set M is defined by a tuple (S_i, s, L_i, N_i, t_i), where

8

1. S_i is a finite set of states.

2. s, an element of S_i, is a designated **state** called the **initial state** of m_i.

3. L_i is a finite set of **local variables**, with a specified name and a finite range.

4. N_i is a finite set of names, each of which is associated with a unique pair (p,a), where "p" is a **predicate** on the variables $L_i \cup R_i$, and "a" is an **action** on the variables $L_i \cup R_i \cup W_i$. An action is a partial function from the values contained in the local variables and read access variables to the values of the local variables and write access variables:

$$a: L_i \times R_i \longrightarrow L_i \times W_i.$$

5. t_i is a partial transition function from the states and names of m_i to the states of m_i:

$$t_i: S_i \times N_i \longrightarrow S_i$$

The machines model the entities of a system, which may be processes, channels, subsystems or stand alone computers. The shared variables are the means of communications between machines. Shared variables are an abstraction of the communications medium across which two machines exchange messages and data. The read access variables (R_i) and the write access variables (W_i) are subsets of the set of all variables (V) to which machines (m_i) have access. The read access variables are used by individual machines to determine which enabling predicates are true. A machine may make a transition (t_i) from one state to another when the enabling predicate associated with the name for that transition (N_i) is

9

true. Upon executing a transition, the action associated with the name is executed. The action may change the values of both local and shared write access variables, thus enabling other predicates. The execution of a transition may be considered an atomic action, in which both the state change and the action associated with it occur simultaneously.

The status of a system of communicating machines is characterized by a system state tuple, a system state and a global state. The **system state tuple** lists the current state of each machine in the system. For example, (M,V) is a system of n communicating machines. The state of a machine m_i is labeled s_i, for $0 <= s <=$ the maximum number of states for that machine. The n-tuple $(s_1, s_2, \ldots, s_n)$ is the system state tuple of (M,V). A **system state** is a system state tuple, plus the outgoing transitions which are currently enabled. Two system states are equivalent if every machine is in the same state, and the same outgoing transitions are enabled. The initial **system state** is the system state such that every machine is in its initial state and the outgoing transitions are those enabled from the initial state.

The **global state** of a system consists of the system state plus the values of all variables both local and shared. It may be written as a larger tuple, combining the system state with the values of the variables. The **initial global state** is the initial system state with the additional requirement that all variables have their initial values. A global state

10

corresponds to a system state if every machine is in the same state, and the same outgoing transitions are enabled.

B. USE OF THE MODEL FOR ANALYSIS

System states and global states may be utilized to conduct a reachability analysis of a system of communicating machines and thus determine if the model is free from certain types of errors. Three types of errors, deadlock, unspecified receptions and nonexecutable transitions may be identified through reachability analysis. Deadlock occurs when all machines are unable to progress. SYSCOM defines a **deadlock** as a system state in which every machine m_i is in a state X_i, such that no transition out of state X_i is enabled. An **unspecified reception** occurs when a message is received by a machine through a communications channel and the machine for which it was intended is unable to receive it. Finally a **nonexecutable transition** is a specified transition which can never be executed from the initial system state.

System states and global states are utilized to conduct a reachability analysis of a system of machines by exhaustive analysis of machine states, local and shared variables and all possible transitions. If the values of all variables are restricted to a finite range then the system may be reduced to a set of finite states. If the variables may take on a wide range of values or are not restricted to finite range then the number of global states may be very large or infinite,

11

potentially preventing a reachability analysis. However, even
if the number of global states is infinite, the number of
system states is still finite because the number of states
defined in each machine (S_i) is finite. Thus a reachability
analysis may be conducted on the system states even though the
number of global states may be infinite. Herein lies the
advantage of **System of Communicating Machines**, there is
potential for a large reduction in the total number of states,
as compared to conventional finite state machine models, which
in turn significantly reduces the size of the reachability
graph and is still sufficient to determine if many protocols
are error free.

III. SPECIFICATION OF A SIMPLIFIED MIL-STANDARD 1553

A. SIMPLIFICATIONS AND ASSUMPTIONS

The 1553 Bus protocol may be expressed utilizing the System of Communicating Machines model. A simplified version is presented here, but more complete versions may be generated from this basic model. This version describes normal command/response communication between a Bus Controller and multiple Remote Terminals.

The following simplifications have been incorporated into this model:

1. No timing is modeled. It is assumed that transitions specified occur within the time limits delineated in the standard.

2. The transmission medium (bus) is free of errors.

3. None of the optional features, such as the "busy bit," are included.

4. The broadcast mode of operation is not implemented.

5. Terminals operating as a Bus Monitor are not implemented.

6. The Command, Status and Data Words described in the Mil-Spec have been simplified and some fields combined for ease of explanation.

7. Smart remote terminals have been modeled which are actively involved in monitoring terminal to terminal transfers. [Ref. 4:pp. 1-6] refers to RTs validating addresses.

The primary purpose of these simplifications is to promote a basic understanding of how the standard is intended to

operate in a normal mode. Once the operation of the basic model is clear each simplification may be incorporated into the model to make it correspond directly with the complete Mil-Standard.

B. EXPRESSING MIL-STD 1553 AS A SYSTEM OF COMMUNICATING MACHINES

The Mil-Standard 1553 is formally represented as an ordered pair, $C = (M, V)$, where

$$M = \{M_{BC}, M_{RT}(i) \text{ for } 0 <= i <= 31\}$$

is a finite set of machines and

$$V_{1553} = R_{1553} = W_{1553} = \{COMMAND1, COMMAND2, STATUS, DATA\}$$

is a finite set of shared variables. The read access variables and write access variables are equivalent. Thus, each one of the four variables may be read from or written to by the bus controller or any remote terminal. Figure 3 represents the bus controller, remote terminals exchanging data via the shared variables. The bus controller is defined as follows:

$$M_{BC} = \{S_{BC}, s, L_{BC}, N_{BC}, t_{BC}\} \text{ where,}$$

$$S_{BC} = \{0,1,2,3,4,5\} \text{ and}$$
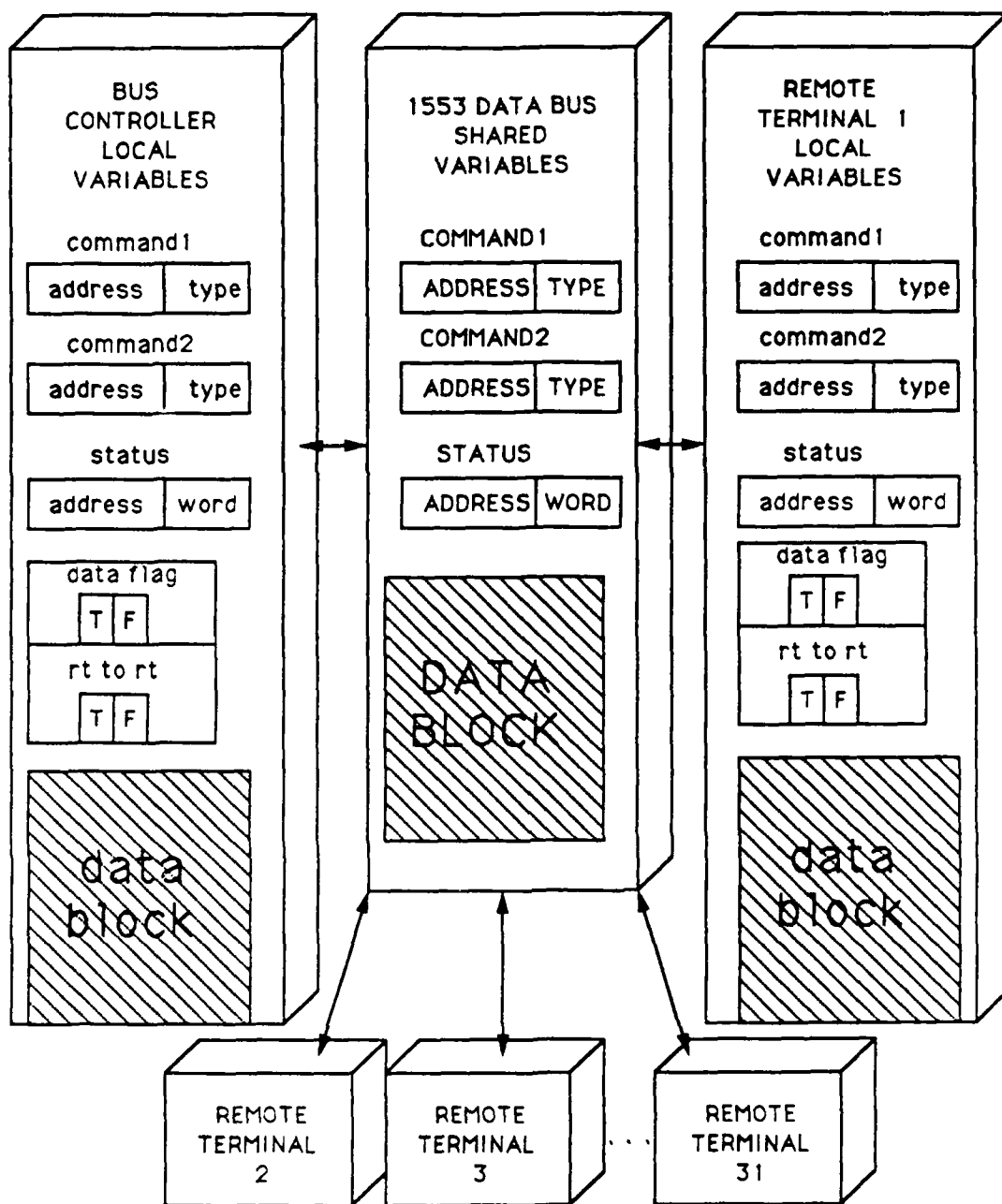
Figure 3. Mil-Standard 1553 System of Machines

$$s = 0 \text{ and}$$

$$L_{BC} = \{\text{command1, command2, status, data, rt\_to\_rt}\}.$$

N_{BC} describes the names of all state transitions with associated enabling predicate and required action. The finite state machine for the bus controller, t_{BC}, is represented in Figure 4. The complete predicate action table is depicted in Table 2.

Remote terminals are defined in similar fashion. Each remote terminal is represented by a similar tuple as follows:

$$M_{RT} = \{S_{RT}, s, L_{RT}, N_{RT}, t_{RT}\} \text{ where,}$$

$$S_{RT} = \{0,1,2,3,4,5,6\} \text{ and}$$

$$s = 0 \text{ and}$$

$$L_{RT} = \{\text{command1, command2, status, data, rt\_to\_rt, dataflag}\}.$$

To complete the definition, t_{RT} and N_{RT} are described in Figure 5 and Table 3 respectively.
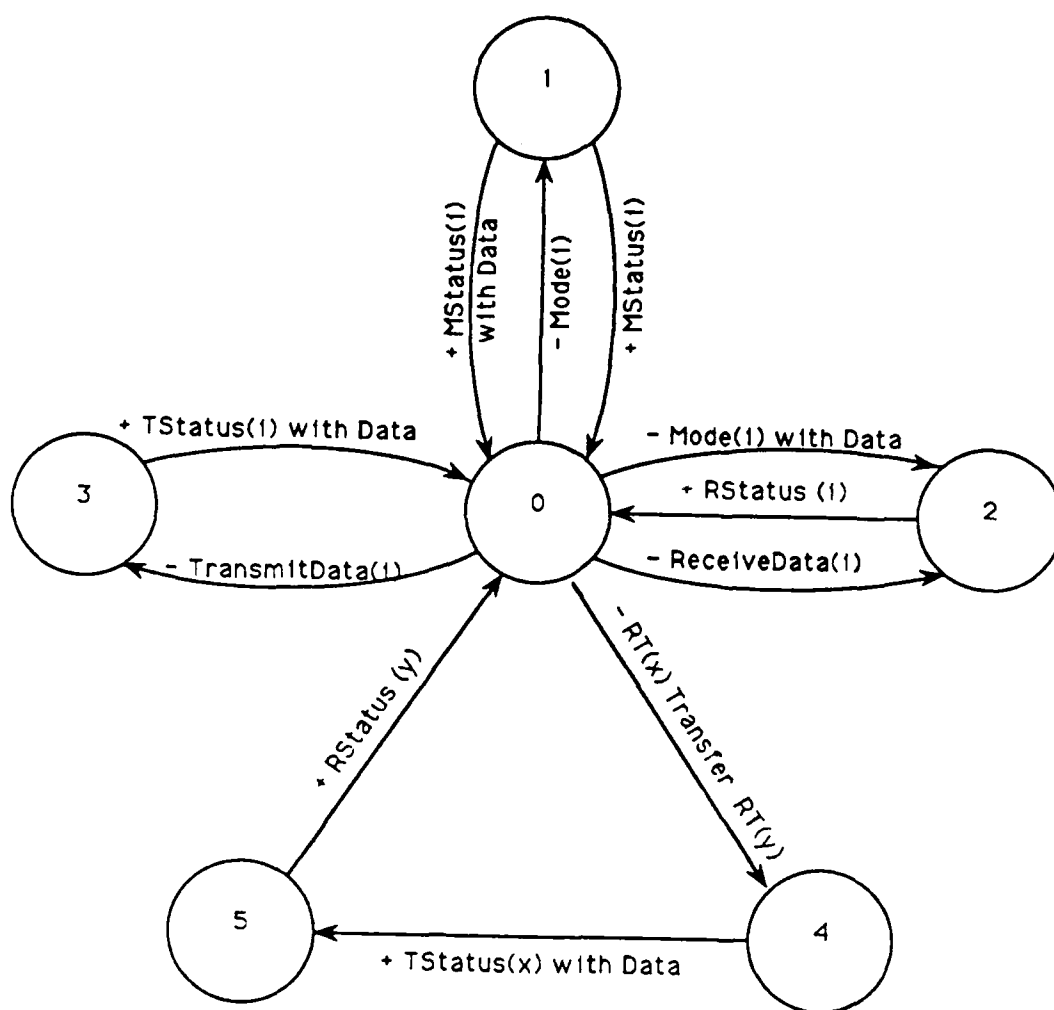
Figure 4. t_{BC} Finite State Machine Diagram for 1553 Bus Controller

TABLE 2

PREDICATE ACTION TABLE FOR BUS CONTROLLER

N_{BC}	PREDICATE	ACTION
- Mode$_i$	command1(addr, type) = RT$_i$, Mode	COMMAND1 := command1
- Mode$_i$ with Data	command1(addr, type) = RT$_i$, Mode with Data	COMMAND1 := command1 DATA := data
- ReceiveData$_i$	command1(addr, type) = RT$_i$, ReceiveData	COMMAND1 := command1 DATA := data
- TransmitData$_i$	command1(addr, type) = RT$_i$, TransmitData$_i$	COMMAND1 := command1
- RT$_x$transferRT$_y$	command1(addr, type) = RT$_y$, ReceiveData and command2(addr, type) = RT$_x$, TransmitData	COMMAND1 := command1 COMMAND2 := command2
+ TStatus$_i$ with Data	command1(addr) = STATUS(ADDR) and DATA <> 0	status := STATUS data := DATA STATUS := 0 DATA := 0
+ TStatus$_x$ with Data	command2(addr) = STATUS(ADDR) and rt to rt = true and DATA <> 0	No Action Monitor only
+ RStatus$_i$	command1(addr,type) = STATUS(ADDR), (Mode with Data or ReceiveData)	status := STATUS STATUS := 0

TABLE 2 (CONTINUED)

N_{BC}	PREDICATE	ACTION
+ RStatus$_y$	command1(addr) = STATUS(ADDR) and rt to rt = true	rt to rt := false
+ MStatus$_1$	command1(addr,type) = STATUS(ADDR), Mode DATA = 0	status := STATUS STATUS := 0
+ MStatus$_1$ with Data	command1(addr,type) = STATUS(ADDR), Mode DATA <> 0	status := STATUS data := DATA STATUS := 0 DATA := 0

19
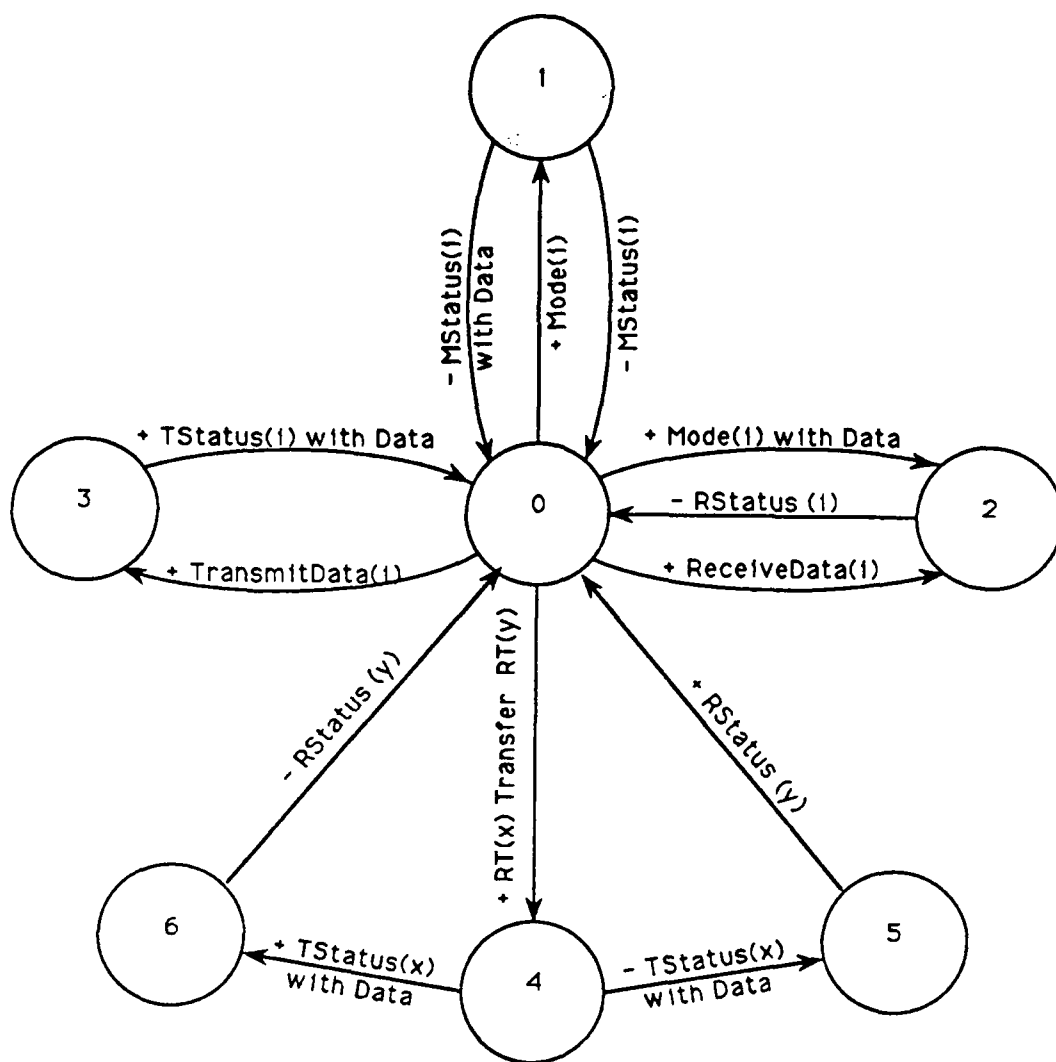
Figure 5. t_{RT} Finite State Machine Diagram
for 1553 Remote Terminal

TABLE 3

PREDICATE ACTION TABLE FOR REMOTE TERMINAL

N_{RT}	PREDICATE	ACTION
+ Mode$_i$	COMMAND1 (ADDR, TYPE) = RT$_i$, Mode	command1 := COMMAND1 if 10000 <= Mode <= 11111 then dataflag := true COMMAND1 := 0
+ Mode$_i$ with Data	COMMAND1 (ADDR, TYPE) = RT$_i$, Mode with Data	command1 := COMMAND1 data := DATA COMMAND1 := 0 DATA := 0
+ RecieveData$_i$	COMMAND1 (ADDR, TYPE) = RT$_i$, Receive Data and COMMAND2 = 0 and DATA <> 0	command1 := COMMAND1 data := DATA COMMAND1 := 0 DATA := 0
+ TransmitData$_i$	COMMAND1 (ADDR, TYPE) = RT$_i$, Transmit Data	command1 := COMMAND1 COMMAND1 := 0
+ RT$_i$transferRT$_y$	COMMAND1 <> 0 and COMMAND2 (ADDR, TYPE) = RT$_i$, Transmit Data	command1 := COMMAND1 command2 := COMMAND2 rt_to_rt := true
+ RT$_x$transferRT$_i$	COMMAND1 (ADDR, TYPE) = RT$_i$, Receive Data and COMMAND2 <> 0	command1 := COMMAND1 command2 := COMMAND2 rt_to_rt := false
+ RStatus$_y$	command1 (addr) = STATUS (ADDR) and rt_to_rt = true	status := STATUS rt_to_rt := false STATUS := 0

21

TABLE 3 (CONTINUED)

N_{RT}	PREDICATE	ACTION

+ TStatus$_x$
with Data
	command1(type) :=
	Receive Data
	    and
	command2(addr) =
	STATUS(ADDR)
	    and
	rt_to_rt = false
		status := STATUS
		data := DATA

- RStatus$_i$	command1(type) = (Mode with Data or Receive Data) and rt_to_rt = false	STATUS := status
- MStatus$_i$	command1(type) = Mode dataflag = true	STATUS := status
- RStatus$_i$	command1(type) = Receive Data and rt_to_rt = true	STATUS := status rt_to_rt := false
- TStatus$_i$ with Data	command1(type) = Transmit Data	STATUS := status DATA := data
- MStatus$_i$ with Data	command1(type) = Mode dataflag = true	STATUS := status DATA := data dataflag := false
- Status$_x$ with Data	command1(type) = Transmit Data and rt_to_rt = true	STATUS := status DATA := data COMMAND1 := 0 COMMAND2 := 0

C. VARIABLES AND PREDICATE ACTION TABLES

The formal specification of the 1553 Bus protocol consists of the set of machines and local variables, the set of shared variables and the corresponding predicate action tables. Some additional explanation may be helpful to aid in understanding the model.

The variables closely parallel the command, status and data word described in the Mil-Standard. This model abstracts the word format to promote understanding of the 1553 protocol. The **COMMAND1** variable may be considered an array, or vector, consisting of **ADDRESS** and **TYPE** fields. The address field corresponds to the five bit address of one of up to 31 remote terminals. The type field contains the type of command which, in actuality, combines the function of the T/R bit, subaddress/mode bits and the data word count/mode code field, all of which are described in more detail in the standard. The **COMMAND2** variable is identical to COMMAND1 and is utilized for terminal to terminal transfers. The **STATUS** variable is a vector consisting of an **ADDRESS** and **WORD** field. The address in the status word always contains the address of the remote terminal transmitting the status word. The word field combines the remaining 17 bits of the status word, each of which has special function. The **DATA** variable is an array consisting of from one to 32 data words. The Mil-Spec requires that the Data Word Count/Mode Code field contained in the command word determine the actual number of data words to

23

be transferred. For this purpose a single data variable is sufficient to model protocol behavior.

Each Machine has its own set of local variables. The local variables are identical in name and composition to the shared variables, but they may only be read from and written to by the machine itself. The bus controller and each RT have an additional local (boolean) variable called **rt_to_rt** which is used to identify when RT to RT transfers are in progress between remote terminals. Each RT has an additional variable called the **dataflag** which indicates when the Bus Controller has directed the Remote Terminal respond to a $Mode_i$ message with status and data words. The Mil-Standard does not include any type of boolean variable.

Figure 3 represents the system of machines. Local variables are contained within the bus controller box and the remote terminal boxes. The shared variables simulating the data bus are contained within the 1553 data bus box. The bus controller and all remote terminals have access to the shared variables of the data bus, as indicated by the bidirectional arrows. The 1553 data bus box represents the physical medium which separates the bus controller from remote terminals.

Tables 2 and 3 represent the predicate action tables for the bus controller and remote terminal. The name of the transition corresponds to a directed edge in the respective finite state machines Figures 4 and 5. The enabling predicate describes the conditions required for the matching transition

24

to be executed. The appropriate actions to be executed are described in the action column. All of the conditions in the enabling predicate must be true in order for a transition to be enabled. Command and status vectors are abbreviated. For example:

COMMAND1.ADDRESS = RT_i and COMMAND1.TYPE = $Mode_i$

is abbreviated as:

COMMAND1(ADDR, TYPE) = RT_i, $Mode_i$

By convention shared variables are capitalized to distinguish them from local variables which are not. For example COMMAND1.ADDRESS is a shared variable while the counterpart, command1.address is local.

D. MESSAGE TYPES AND ASSOCIATED FINITE STATE MACHINES

All communication on the bus is initiated by the bus controller. The bus controller will store one of five general types of messages into the shared COMMAND1 or COMMAND2 variables. The remote terminals will, in turn, access the messages by examining the variables. The bus controller will send the following message types:

1. $Mode_i$--A single command word which directs the remote terminal to assume a specific mode of operation such as assume dynamic bus control or perform a function such as initiate a self test.

2. **Mode$_i$ with Data**--A single command word with a data word which serves the same general purpose as the Mode$_i$ command but is normally utilized for remote terminals which service several subsystems.

3. **Transmit Data$_i$**--A single command word which directs a remote terminal to transmit a specified number of data words back to the bus controller.

4. **Receive Data$_i$**--A single command word followed immediately by a series of data words directed to a remote terminal. The specific remote terminal and the number of data words are contained in the command word.

5. **RT$_x$ transfer to RT$_y$**--A pair of command words which simultaneously directs RT$_x$ to transfer data and RT$_y$ to receive data. The number of data words to be transmitted and received are contained in the command words.

The remote terminals will respond to the commands of the bus controller with either of two types of messages:

1. **RStatus$_i$**--A single status word which indicates successful reception the preceding ReceiveData$_i$ or Mode$_i$ with Data message.

2. **MStatus$_i$**--A single status word which indicates successful reception the preceding Mode$_i$ command.

3. **RStatus$_y$**--A single status word which indicates successful reception the preceding TStatus$_x$ with Data message.

4. **TStatus$_i$ with Data**--A single status word followed by a series of data words which indicates successful reception of and appropriate response to the preceding TransmitData$_i$ message.

5. **TStatus$_y$ with Data**--A single status word followed by a series of data words which indicates successful reception of and appropriate response to the preceding RT$_x$ transfer to RT$_y$ message.

6. **MStatus$_i$ with Data**--A single status word followed by a series of data words which indicates successful reception of and appropriate response to the Mode$_i$ message.

The finite state machine for the bus controller is depicted in Figure 4. Each directed edge of the machine is named and signifies a transition from one state to another during which a message is transmitted or received. The minus (-) sign preceding a message indicates a sending transition and the plus (+) sign a receiving transition. Figure 5 represents a generic remote terminal, RT_i. The edges of the remote terminal complement the bus controller and are labeled in the same manner.

IV. **REACHABILITY ANALYSIS**

In order to test the model to determine if it actually represents the function of the 1553 bus protocol, a directed graph called a reachability analysis is generated. Conventional finite state machines utilize global states which contain the complete state of all machines in the system. Potential exists for a state explosion to occur, in which a very large number of states is generated and the analysis becomes impractical. The analysis of the 1553 model utilizes system states rather than global states which significantly reduces the possibility of state explosion for this and many other types of protocols.

A. STARTING STATE AND SIMPLE MESSAGE AND DATA TRANSFERS

The initial starting state, depicted in Figure 6, finds the bus controller and all remote terminals in state zero. All variables are empty. The rt_to_rt variable for all machines is set to false as is the data flag for each remote terminal. The bus controller initiates communication when its local command and possibly data variables are loaded with a message. A transition will be enabled, based upon the contents of the local variables corresponding to the requirements in the predicate action table. The bus controller will write to the shared variables which in turn
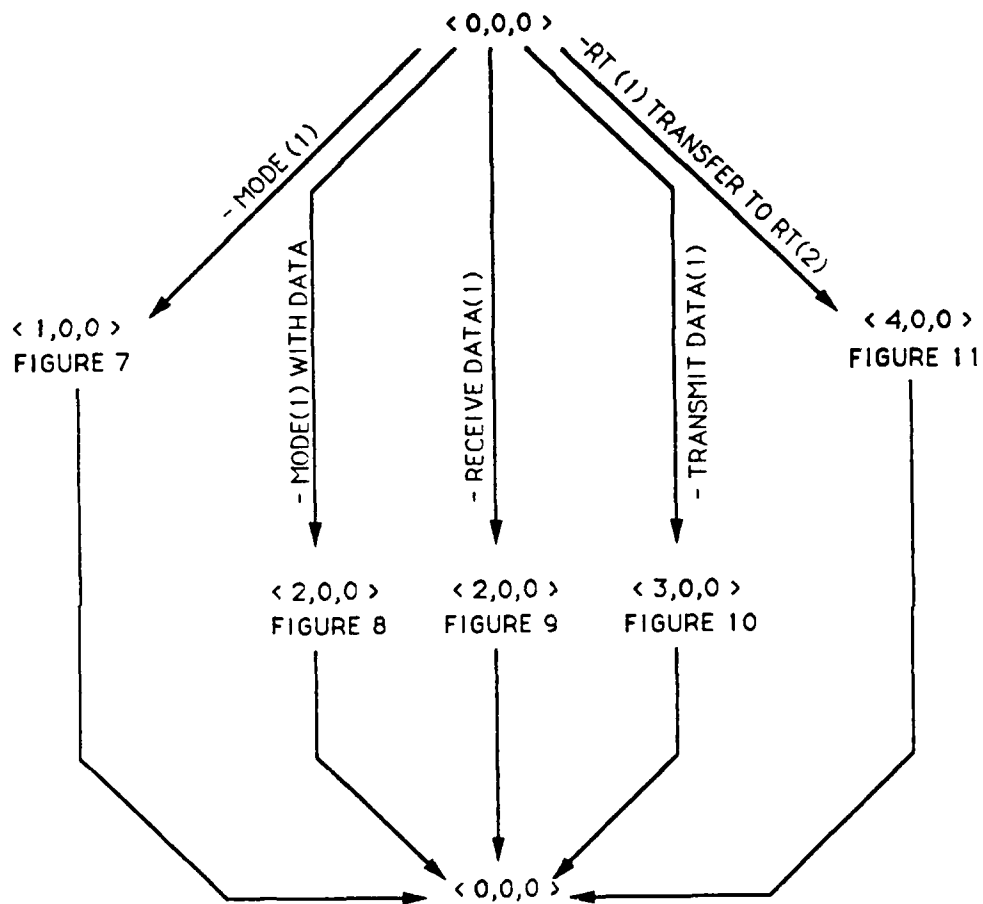
Figure 6. Reachability Analysis of Mil-Standard 1553

may be accessed by all remote terminals. Only remote terminals for which enabling predicates are satisfied may make subsequent transitions and if applicable clear the shared variables. The remote terminal interprets the type of command and writes back to the shared variables. The bus controller in turn reads the shared variables and clears them to complete the cycle. Figures 7, 8, 9, and 10 represent the normal sequence of states transitions which occur for $Mode_i$, $Mode_i$ with Data, $ReceiveData_i$ and $TransmitData_i$ messages.

B.   TERMINAL TO TERMINAL TRANSFERS AND POTENTIAL DEADLOCKS

In the case of terminal to terminal transfers, both remote terminals and the bus controller monitor the shared variables throughout the exchange of information. This redundancy is introduced in order to ensure that data is both transmitted and received properly by the RT's. Although not directed by the Mil Standard, this models a more robust system with intelligent remote terminals. Figure 11 presents the more complex sequence of states which occurs during an RT_x transfer to RT_y message.

Three potential deadlocks exist which, unless explained, indicate that the standard is in error. The first deadlock occurs in system state <4,5,0>, when RT_1 attempts to transmit a $TStatus_1$ with Data message to RT_2. RT_1 could transmit its message before RT_2 has received the RT_1 transfer to RT_2 message. In actuality this is simply a timing problem, that will not

30

FROM FIGURE 6

‹ 1,0,0 ›

+ MODE(1)

‹ 1,1,0 ›

- MSTATUS(1) WITH DATA

- MSTATUS(1)

‹ 1,0,0 ›

‹ 1,0,0 ›

+ MSTATUS(1) WITH DATA

+ MSTATUS(1)
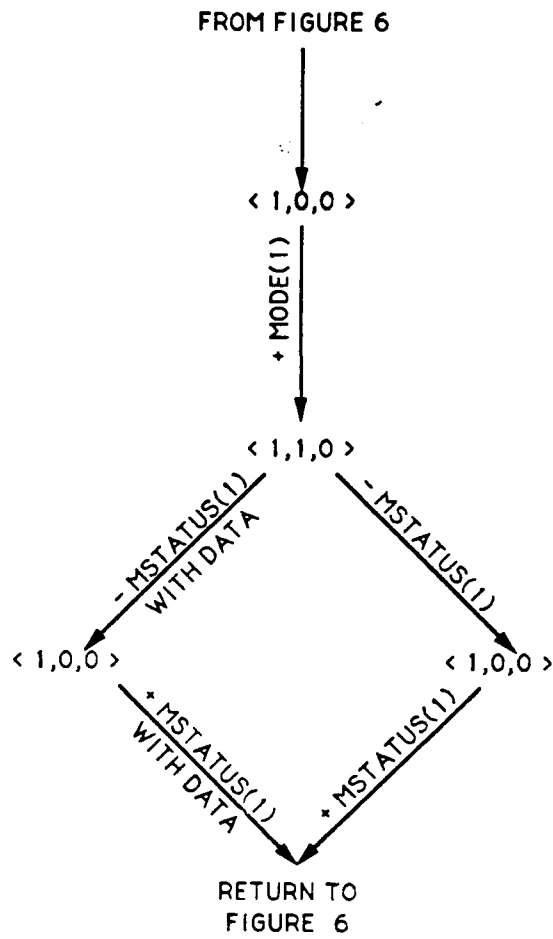
RETURN TO
FIGURE 6

Figure 7.   Reachability Analysis for Mode(1) Command

31

FROM FIGURE 6

↓

< 2,0,0 >

+ MODE (1)
WITH DATA

↓

< 2,2,0 >

− RSTATUS(1)

↓

< 2,0,0 >

+ RSTATUS(1)

↓

RETURN TO
FIGURE 6

Figure 8. Reachability Analysis for
Mode(1) with Data Command

FROM
FIGURE 6

‹ 2,0,0 ›

＋ RECEIVE
DATA(1)

‹ 2,2,0 ›

－ RSTATUS(1)

‹ 2,0,0 ›

－ RSTATUS(1)
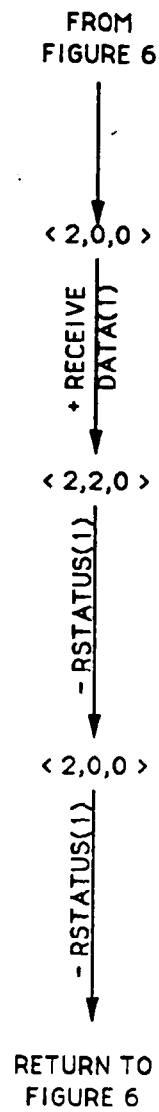
RETURN TO
FIGURE 6

Figure 9. Reachability Analysis for
Receive Data(1) Command

33

FROM
FIGURE 6

⟨ 3,0,0 ⟩

+ TRANSMIT
DATA(1)

⟨ 3,3,0 ⟩

− TSTATUS(1)
WITH DATA

⟨ 3,0,0 ⟩

− TSTATUS(1)
WITH DATA

RETURN TO
FIGURE 6

Figure 10.   Reachability Analysis for
Transmit Data(1) Command

34
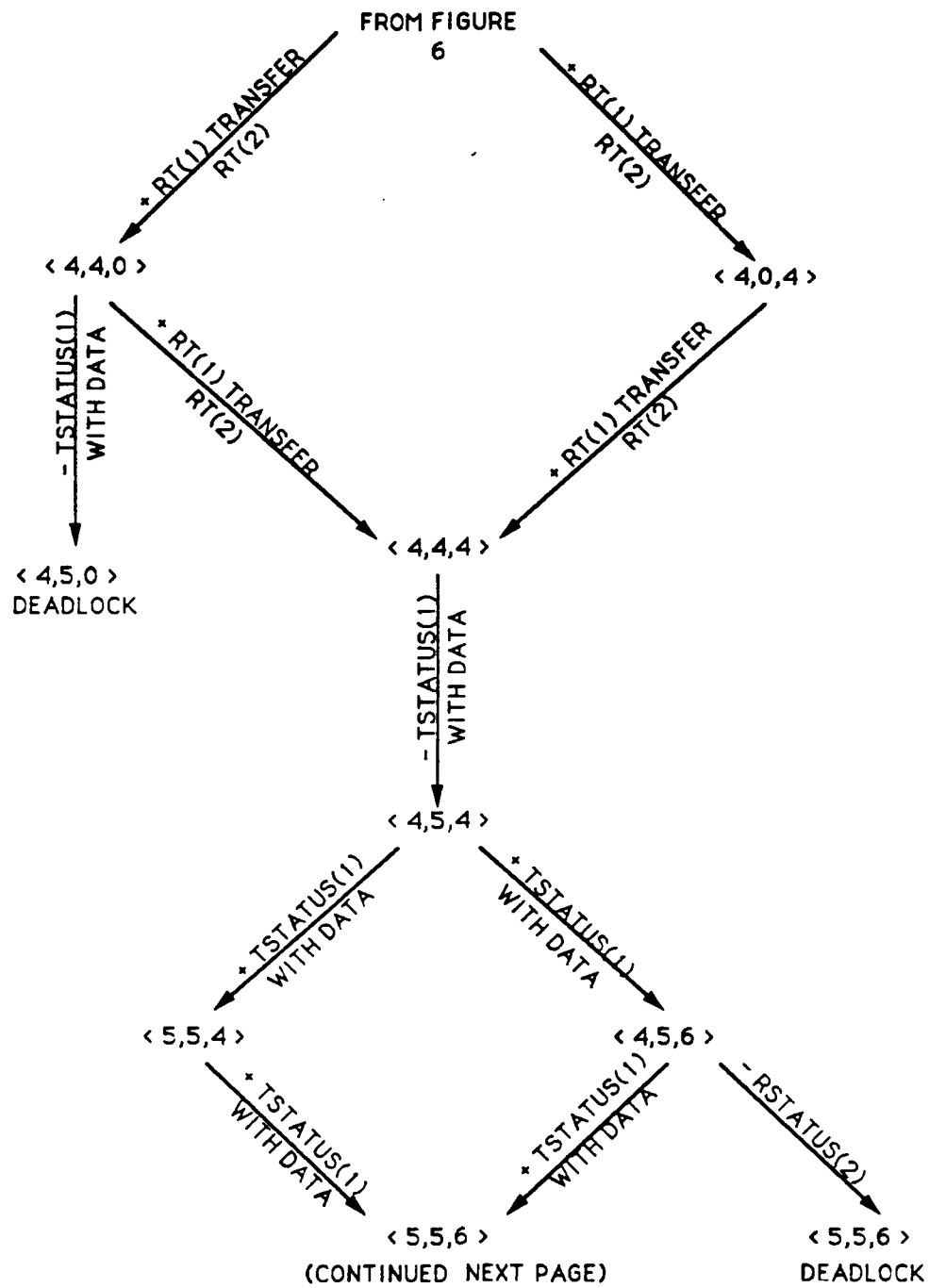
Figure 11. Reachability Analysis for RT(1)
Transfer to RT(2) Command

CONTINUED FROM
PREVIOUS PAGE

< 5,5,6 >

-RSTATUS(2)

< 5,5,0 >

*RSTATUS(2)

*RSTATUS(2)

< 0,5,0>
DEADLOCK

< 5,0,0 >

+ RSTATUS(2)

RETURN TO FIGURE 6
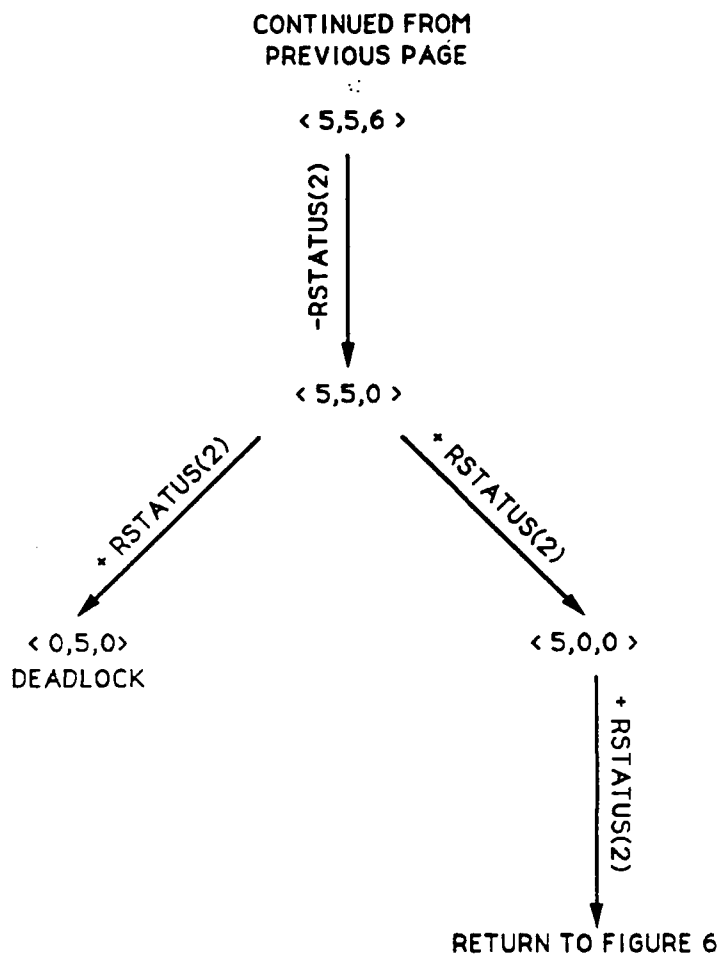
Figure 11.   (CONTINUED)

occur in the real world because RT_1 would not transmit its message to RT_2 before the minimum response time (4 usec) had elapsed.

The second deadlock occurs when RT_2 attempts to send a $RStatus_2$ message to the bus controller, signifying the successful reception of data. In this instance the bus controller in not "ready" to receive the status word, because it has not yet executed the "+ $TStatus_1$ with Data" transition.

The final problem area occurs in system state <0,5,0> in which the bus controller clears the shared STATUS variable before remote terminal RT_1 has complete reading the variable. This deadlock occurs because the limitations of the model in modeling the physical medium of the data bus. In actuality, this deadlock represented by the model cannot occur, because the bus controller and remote terminals actually execute receiving transitions virtually simultaneously. That is, both machines receive the signal on the bus, which is modeled by a shared variable.

V.   GOING BEYOND THE SIMPLE MODEL

Systems of Communicating Machines can be used effectively to model a simplified version of the Mil-Standard 1553 Bus Protocol, and demonstrate viability. Bus timing requirements, the broadcast mode of operation, errors and busy bit may all be incorporated by adding to the predicate action tables and finite state machines. The following examples will show how each feature may be added.

The 1553 standard has strict timing requirements which must be adhered to by both the bus controller and remote terminals. The bus controller must provide a minimum time period of four usec, called the intermessage gap, which falls between the completion of one series of messages and the beginning of the next. Remote terminals must respond to valid messages within a four to 12 usec response period which follows receipt of a message. The bus controller may supersede a valid command until the response period commences four usec after transmission of the initial message. If the remote terminal is unable to respond within 12 usec then it does not respond at all. The bus controller will wait for up to 14 usec for a response from the remote terminal before time out occurs.

A. MODELING TIMING AND TIMEOUTS

In order to incorporate timing, changes are required to the enabling predicates of both the bus controller and remote terminal and additional transitions must be added to the finite state diagrams. The bus controller and remote terminal require an additional local variable called **event timer**. This variable takes on integer values and an increment action is continuously enabled. It is assumed that the variable is incremented at one usec intervals for this protocol. The following changes may be incorporated into the predicate action table:

1. The enabling predicates for sending transitions for the bus controller would permit the command words in the shared variables to be overwritten by the bus controller. The overwrite may occur prior to the local event timer variable indicating four usecs. Only after four usec had elapsed would the appropriate actions and state transitions associated with the last command word be permitted. The enabling predicate would appear as follows:

    local event timer >= 4 usec.

2. The enabling predicates for receiving transitions for the remote terminal would require the shared variables be monitored for changes for the entire four usec period after receiving the first message. After four usec had elapsed, as indicated by the bus timer variable, appropriate action may be taken and transition effected, with the same enabling predicate mentioned above.

3. The remote terminal could only execute sending transitions during the four to 12 usec response period. For example:

    4 usec <= local event timer <= 12 usec.

4. The bus controller will await a response from a remote terminal for up to 14 usec. If the local event timer was greater than 14 usec then the enabling predicate for any receiving transition for bus controller and any sending transition and some receiving transitions for remote terminals must be disabled. In addition a new transition is required which would enable all machines to revert from their existing state to state zero, and clear all shared variables. The new transition, labeled **timeout**, would be enabled as follows:

local event timer >= 14 usec.

Thus, timing may be incorporated by simply adding to the predicate action tables of the bus controller and remote terminals and adding a new transition to the finite state machines, and modeling the clock with a variable.

B. MODELING ERRORS

Two types of errors may occur when a message is transmitted from one machine to another. Errors in data reception may be caused by a physical error in the format of the transmitted command, status or data words, which permits the message to be received, but prevents appropriate response. Invalid data reception may also result from an error in the number of data words sent from one terminal to another.

Modeling these errors involves further changes to the predicate action tables, but no additional state transitions. The bus controller and remote terminal will respond similarly when a message is received with errors. When the bus controller receives a message with errors from a remote terminal, the message must be ignored and the machine must

40

revert to state zero to reinitiate an exchange. When a remote terminal receives a message with an error, it must be inhibited from responding to the message, and subsequently time out. Predicates and actions must be modified as follows:

1. The status variable must have an additional element added to the vector called the **message error** bit, which indicates when an error has been detected. A local variable called **message error** must be added to the bus controller and each remote terminal. This message error variable must be set true when an error is detected. The timeout transition is renamed **timeout/error**, to more accurately describe the function of the transition.

2. When the bus controller or remote terminal detects an error during a receiving transition, that machine is inhibited from making subsequent sending transitions. The machine which detects the error will, instead, make the timeout/error transition whenever the message error variable is set true. The enabling predicate for all sending transitions is modified to include:

message error = false.

The enabling predicate for the timeout/error transition is modified to include:

message error = true.

Thus, the method proposed here to manage errors will prevent deadlock, by permitting each machine to revert to state zero. The machine receiving the error is in effect forced along the timeout/error transition by the message error variable. The machine transmitting the error, regardless of the state of the receiving machine, is forced to default passively via the same transition to the initial state due to the resultant timeout.

41

C. THE BROADCAST MODE

The broadcast mode of operation is used when one machine has information to transmit to all other machines listening on the bus. All remote terminals set a bit in their status word if the broadcast command was received properly, but the status word is not transmitted back to the bus controller. This mode of operation is described in the standard as a "significant departure from the basic philosophy of this standard in that it is a message format which does not provide positive closed loop control of bus traffic."

The formal specification must be modified to include a common broadcast address, new predicates and actions and new transitions. The unique address 11111_2 identifies specific commands as broadcast commands. Only the following types of messages may be transmitted to support the broadcast mode:

1. ReceiveData$_{All}$--The bus controller sends a single command word followed by a contiguous series of data words to all remote terminals.

2. RT_x transfer to RT_{ALL}--The bus controller commands RT_x to transmit a status word followed by a series of data words to all RTs.

3. Mode$_{All}$--Bus Controller sends a single command word to all remote terminals directing a specific mode of operation.

4. Mode$_{All}$ with Data--Bus controller sends a single command word with data word to all remote terminals directing a specific mode of operation.

The sending transitions from the bus controller and receiving transitions for the remote terminals need not be modified. The enabling predicates and associated actions for

42

bus controller sending and remote terminal receiving transitions are similar to those associated with the analogous messages of the normal mode of operation. The enabling predicates must be modified to recognize the common broadcast address and a local boolean variable **broadcast command received** must be added the status variable vector. Finally a broadcast command received bit must be added to the status vector. Remote terminals may use the boolean variable to identify when a broadcast command has been received. A null transition from the receiving state back to state zero must be added to remote terminal finite state machines to inhibit subsequent transmission of multiple status words simultaneously. Predicate action tables will write the status of the broadcast command received bit into the local status vector. Handling of errors will not be affected by addition of this new mode of operation.

D. METHODOLOGY FOR BUILDING UPON THE BASIC MODEL

The following methodology may be utilized to add features to this or any other SYSCOM Model.

1. Decide which new feature is to be added to the existing model.

2. Determine if the new feature is local to each machine or global to the system of machines or both.

3. Determine if modification of the existing local/global variables is sufficient to represent the new feature or if new variables are required.

4. Determine if the new feature will require a new transition for the Finite State Machines.

43

5. Make modifications to Predicate/Action Table.

6. Conduct Reachability Analysis to test validity of modifications.

The simple methodology proposed here will permit development of a more complete model which may be directly converted into software, firmware or hardware.

VI. TRANSPARENT INTEGRATION OF THE ASQ-191 INTO THE EA-6

A. PURPOSE AND BACKGROUND

The focus of this section is to define the requirements and propose an approach to integrating the AN/ASQ-191 Radio Countermeasures Set into the EA-6B Improved Capabilities (ICAP II) 1553A bus architecture. The requirements to integrate into the P-99 and Block 86 configurations of the EA-6B are presumed to be similar. 1553 bus architecture is based upon the coherent transmission of Command, Status and Data words between a Bus Controller and Remote Terminals. A methodical approach to software design is required in order to effect this coherent exchange. Effective software engineering utilizes a top-down approach beginning with a statement of requirements.

The EA-6B is a carrier-based tactical aircraft solely dedicated to Electronic Warfare. The ALQ-99 Tactical Jamming System is the heart of the EA-6B and has demonstrated excellent capability in jamming conventional radar systems but has limited ability to collect against, and counter Command/ Control and Communications (C3) threats. In response to lower frequency threats, the early versions of the EA-6B were equipped with the Vietnam era ALQ-92 which was manually operated and unreliable. More recently the EA-6B has been equipped with the ASQ-191. [Ref. 5:p. 1] describes the

45

systems capabilities which include smart ESM and ECM against communications threats.

Although the ASQ-191 significantly enhances EA-6B capabilities, the system is not integrated into the ALQ-99 system. In addition the control unit for the ASQ-191 is located in the front cockpit, whereas an operator, located in the aft cockpit, in actuality will make employment decisions and operate the ASQ-191. Some specific areas for concern include the following:

1. ECMO (Electronic Countermeasures Officer) #1 is required to operate and monitor the ASQ-191 from the front cockpit which detracts from his other responsibilities as Co-Pilot and Navigator.

2. Aft cockpit system operators are required to prompt ECMO #1 for information and requests to change operating parameters for the ASQ-191. Confusion and errors may result and increased intercockpit communications are required.

3. Opportunities for employment of the ASQ-191 may be lost due to the physical displacement of the unit from the primary operators, which in turn results in reduced combat effectiveness.

Integration of the ASQ-191 into the 1553 data bus would alleviate the areas of concern above. In addition the ASQ-191 is the interim to the ADVCAP (Advanced Capability) EA-6B/ALQ-149, which is not planned for introduction into the fleet until the mid 1990's. With the ASQ-191 serving as an interim Command Control and Communications Countermeasures (C3CM) capability, issues relating to rapid intercept and automated response to communications threats must be resolved in the near term rather than deferred. TEAMS databases should be

applied to the ASQ-191 and C3CM tactics developed in advance of ADVCAP Initial Operational Capability (IOC).

Transparent integration of the ASQ-191 into the EA-6B will call for changes to the ASQ-191, ALQ-99, and AN/TSQ-142 Tactical EA-6B Mission Support System (TEAMS). The impact upon the ALQ-99 Tactical Computer and Display processor programs should be minimal. Limited excess memory in both the Tactical Computer (AYK-14) and Display Processor (ASN-123) coupled with difficulty in modifying firmware in the ASN-123 are important, but not insurmountable considerations.

The following sections will describe the ASQ-191 and propose requirements for changes to the ASQ-191, ALQ-99 and TEAMS to support transparent integration, an established operational requirement in numerous EA-6B deployments with the ASQ-191.

B. AN/ASQ-191 DESCRIPTION

The ASQ-191 components and operations are described in [Ref 5:p. 1-68]. The Software Requirements Specification for the System Controller [Ref. 6] provides additional information including block diagrams. Figure 12 is a system block diagram similar to that found in [Ref. 6:p. 4]. The System Controller is central to the system which interfaces with the Cockpit Control, Data Loader Unit, Receiver/Exciter and High Power Amplifier. [Ref. 6] addresses the interface requirements between the controller and peripheral devices.

47

Figure 12. ASQ-191 System Components

The System Controller provides for complete control of the system through algorithms stored in firmware. Frequency tables and operating parameters are stored in non-volatile memory. Frequency Tables are used to command the Receiver/ Exciter to search for, and jam signals. Operating parameters such as scanning formats, scanning modes and special functions are used to invoke specific firmware algorithms.

Frequency Tables and operating parameters may be loaded via the Data Loader unit and changed via the operator control. ASQ-191 command and status parameters are transmitted to/from the controller in two 8-bit bytes. Frequency Tables, also referred to as Fill data, are transferred in hexadecimal format blocks.

The Cockpit Controller permits complete control of ASQ-191 operating parameters. The Operator control consists of a display, keypad and switches and interfaces to the Data Loader, optional printer and System Controller. Normal interface between the operator and the ASQ-191 is via the Cockpit Control.

Preliminary requirements to integrate the ASQ-191 into the 1553 data bus were developed by the EA-6B Systems Division at the Pacific Missile Test Center, Pt. Mugu California. [Ref. 7] describes a proposal to integrate the ASQ-191 into the 1553 bus in conjunction with a Communications/Radar Exciter [CRE] currently under development. Due to the potential high payoff of integrating the ASQ-191 into the bus structure in

conjunction with CRE development, NAVAIRSYSCOM (PMA-234) is considering the proposal. However, [Ref. 7] is comprehensive and complete, given that information available regarding the ASQ-191 software and hardware configuration was limited, and will be the template upon which additional hardware and software requirements are forwarded.

C. SYSTEM FUNCTION

System function would be based upon the same sequence of events that a normal mission would require: TEAMS mission planning, ALQ-99/ASQ-191 initialization and normal in-flight operations. The following briefly describes anticipated normal operations.

Mission planning utilizing TEAMS would permit identification of both Radar and Communications threats. TEAMS should be used to generate tables for ASQ-191 operation and libraries for ALQ-99 display. TEAMS would permit preplanning of the ASQ-191 system initialization parameters and would maintain a database of optimized communication jamming techniques. Both ASQ-191 and ALQ-99 mission data would be stored on a mission (RRS) tape for future loading into the ALQ-99 and ASQ-191. Post Mission analysis of ASQ-191 intercepts stored on RRS tapes will also be available through TEAMS.

Upon ALQ-99 and ASQ-191 initialization, communications between AYK-14 Tactical Computer and the ASQ-191 System

Controller will be established across a 1553 bus interface. The ALQ-99 will provide system initialization data to the ASQ-191. Prior to the mission, ALQ-99 system operators will load the RRS tape and program the ALQ-99 and ASQ-191 systems to search specific frequency ranges and discrete frequencies. Loading TEAMS generated ASQ-191 Target Tables across the 1553 interface will be similar to loading ALQ-99 emitter libraries. Thus, the ALQ-99 will serve as an alternate to the existing Data Loader Unit for the ASQ-191 Controller.

During normal operations, the ASQ-191 will appear as a supplemental ALQ-99 receiver. At the same time the ALQ-99 will function as a Remote ASQ-191 Cockpit Controller. Both the ALQ-99 and the ASQ-191 systems will require software modules which serve as protocol converters to permit translation of data and commands. Protocol converters would pass information down to, and receive information up from the lower level 1553 interface.

The ASQ-191 will be controlled similar to the existing ALQ-99 receivers. Commonality of operation with the existing ALQ-99 system simplifies workload for, and makes integration transparent to the operator. For example, when the ASQ-191 intercepts a signal, the signal parameters will be passed to the ALQ-99 across the 1553 interface. Signal data should be passed in a format compatible with the ALQ-99 and displayed similar to existing Alarm Words. The operator should be permitted to slew to the ASQ-191 alarm, resume scan, modify

51

target tables and load new target tables, just as he would an ALQ-99 receiver. The command to slew the receiver would be converted into an ASQ-191 "Hold" operation which would be transmitted across the 1553 interface to the ASQ-191 controller.

ASQ-191 Jamming will not correspond to existing ALQ-99 jamming assignments. No changes to ASQ-191 Jamming Modes and modulations are specified. Initial ASQ-191 ECM/ESM modes will be preplanned on TEAMS and may be modified during the mission from the aft cockpit. Software should also permit the J/M switch to be toggled from the aft cockpit. When jamming is initiated the ALQ-99 should generate jammer boxes continuously around the time shared jammed discrete frequencies.

In order to operate the ASQ-191 in optional COMM-1 and COMM-2 modes r: operation (communications, non-C3CM), the ASQ-191 should be placed under local (front cockpit) control only. The intent is to minimize additional software changes and workload for the ALQ-99 Tactical Computer.

D. ASQ-191 HARDWARE AND SOFTWARE CONSIDERATIONS

    1. ASQ-191 Hardware

        The first consideration for ASQ-191 integration is that of hardware. Note that there is currently no mention of, or provisions for, a Mil-Standard 1553 Bus interface for the ASQ-191. However, [Ref. 6:p. 14] shows the following requirement for remote control: "The System Controller shall

be capable of being fully controlled from an RS-232 remote interface. This interface is also used for the Data Loader." The remote RS-232 interface is not equivalent to a 1553 interface. The RS-232 data rate is significantly slower at 19.2 kilobits than the 1553 interface at one megabit. The one megabit data rate of the 1553 interface also exceeds the 250 kilobit data rate of the "high speed" local interface.

Many commercial Remote Terminal Cards are available which could potentially replace the existing remote RS-232 port in the System Controller and Data Loader unit. (Because the Data Loader uses the remote interface, both the System Controller and the Data Loader may require modification.) Existing 1553 bus lines running in proximity to the ASQ-191 Controller and Data Loader could be used to tie into the bus.

Additional software requirements go along with the addition of the 1553 interface. Recent technical review of the ASQ-191 program indicates there exists a 100% reserve in both EPROM and RAM in the System Controller and Cockpit Control. In addition, the Data Loader has a 100% EPROM reserve and a 400% RAM reserve.

Given the reserve software capacity and existing provisions for external control of the ASQ-191 incorporation of a 1553 capability appears feasible.

[Ref. 7:p. 2] indicates that significant concern centers upon the placement of antennas and effects of low-band jamming and potential blanking requirements. Operational

53

deployments have established no EMI problems, which precludes requirement for an intersystem-blanking scheme. No hardware-blanking schemes, antenna placement modifications or other major hardware modifications are necessary for the transparent integration of the EA-6B/ALQ-99 and ASQ-191.

2. ASQ-191 Software Change Requirements

[Ref. 8] provides a complete description of the software design, methodology and function of the ASQ-191 System Controller. [Ref. 8:p. 14] indicates that the ASQ-191 System Controller software falls into two categories, background processing and interrupt processing. An executive routine controls the background processes. Interrupt routines are invoked based upon system configuration and operating mode. [Ref. 8:p. 15] represents the hierarchical organization of the background processes.

The following modifications are proposed to supplement or change selected features/modules in the System Controller, Cockpit Controller and Data Loader Unit.

a. System Initialization

System initialization is normally performed when the ASQ-191 is powered up or restarted. The routine initializes the system controller based upon default parameters, switch settings and fill data in nonvolatile memory.

Some changes are anticipated during initialization to rapidly establish communications between the ALQ-99 and

ASQ-191 and permit the TEAMS generated initialization data and ECM/ESM Tables to be loaded into the ASQ-191 System controller.

The initialization process should attempt to establish communications over the remote 1553 interface prior to permitting local cockpit control. If the ASQ-191 is unable to establish 1553 communications then initialization will default to existing procedures. When the 1553 interface is established the ASQ-191 initialization data stored in the ALQ-99 should be loaded. (ASQ-191 Initialization Data will be generated at a TEAMS workstation and may be modified through the ALQ-99.) Normal and Priority Target Tables and Limited Search Tables will be loaded based upon Library information developed at a TEAMS workstation. It is assumed that a target table loaded across the 1553 interface will normally be entered into nonvolatile memory as Table #1. Other fill data for COMM-1 and COMM-2 modes should remain unchanged. Time of day clock should be synchronized with the AYK-14.

The overall goal is to make communications with the ALQ-99 across the 1553 bus the preferred mode of operation and force default settings to correspond with preflight mission planning.

b. Remote Local Logic

The ASQ-191 is currently configured to select between local "high speed" interface and remote (1553) interface, based upon the remote/local input. Changes to this

logic are proposed to prevent contention between front and back cockpit, to permit operators to swap control of the ASQ-191 and allow the system to gracefully degrade if the 1553 interface is lost due to some type of failure.

The ASQ-191C does provide a remote/local switch on the Cockpit Controller. Changes to the software in the Cockpit controller are required to permit the ASQ-191 to transfer remote control to the ALQ-99. Provisions must also be made to permit the remote local input to be toggled with software from the ALQ-99. If the remote/local input is set for local control then the Cockpit Control should be the only source of operator input. If the input is set for remote control then the remote 1553 interface should be the primary source of input. If 1553 bus communications are not established or fail, then the ASQ-191 should revert to the local interface. This strategy permits either the local, or remote interface to control the system as desired by the operators, prevents contention between multiple operators, and permits graceful degradation should the 1553 interface fail.

Database management problems may occur in switching between remote, local and back to remote interface. Assume the remote interface is used to load target tables and then the system reverts to the local interface. If the target tables are subsequently changed and the remote interface becomes active again, ASQ-191 target tables may not correspond with ALQ-99 active libraries. Cross checking

should be performed to ensure the data previously stored via the remote interface is valid. The goal is to ensure the database is not compromised.

c. Systems Status Update

The current operating status of the ASQ-191 System Controller, Cockpit control and Data Loader Unit must be exchanged regularly to respond to changes in switch positions, operating modes and signal intercepts. The Status Update Software Module in the System Controller currently updates the status of the Cockpit Control and Data Loader Terminals and updates the operator display. Integration into the ALQ-99 System will require additional exchange of status information.

Status update procedures will require some modification to permit a smooth transition between Cockpit Controller and ALQ-99 operation. Inputs should be permitted from only one interface at a time. Changes in system status, made at the ALQ-99 when the 1553 Remote Interface is active, should be sent to the Cockpit Control and Data Loader Unit. For example, currently active Target Tables loaded from the ALQ-99 should be passed for storage in non-volatile memory of both the System Controller and Data Loader Unit. This permits the system to gracefully degrade if the remote interface fails by ensuring that both local Cockpit Control/Data Loader and remote ALQ-99 are operating with the same Data.

The status information that the ALQ-99 requires may be a subset of that required by the Cockpit Controller

57

because the ALQ-99 will not function as a controller for COMM-1 and COMM-2 modes of operation. However, in order to limit the impact upon the ASQ-191 recommend the same status words described in the Voyager Serial Interface Protocols [Ref 8:pp. 71-92] be utilized.

Preliminary examination of the Status Update Module and the Status Words shows how complex the interrelationships are between the System Controller and the Cockpit Control. Close liaison will be required between Rockwell/Collins and PMTC Pt. Mugu to fully identify the requirements to permit the ALQ-99 to receive, interpret and respond to those status words.

d. Built-In-Test and Background Bit

The System controller orchestrates ASQ-191 Built-In-Test (BIT) with a Normal Built-In-Test module. This module sequentially performs a sequence of BITs and outputs results to the operator control. Some simple changes are required to permit initiation of the BIT by the ALQ-99 across the 1553 interface.

The ASQ-191 System Controller should respond to the Mil-Standard 1553 mode command to initiate self-test and respond with results when directed by the Bus Controller. Recommend ASQ-191 BIT be initiated by the ALQ-99 in conjunction with the Onboard System (OBS) Bit. ASQ-191 BIT results shall be displayed so as to limit impact on ALQ-99 displays.

Fault monitoring is also performed in the Normal Built-In-Test module. Results of the fault monitoring must be passed to the ALQ-99 across the 1553 bus. Provisions exist within the ASQ-191 Update Status Module to pass fault-warning messages to the Cockpit Controller. These same fault-warning messages may be passed to the ALQ-99 for display to the aft cockpit operator. Some additional fault-warning messages may be required to provide indications to the front cockpit controller if the 1553 interface fails.

e. COMM-1 and COMM-2 Modes

COMM-1 and COMM-2 modules permit the ASQ-191 to operate as a normal AM/FM Transceiver, and in a special frequency hopping ECCM Mode respectively. Normal C3CM operations would not require system operators to utilize these modes. Because COMM-1 and COMM-2 are supplements to normal communications, propose these modes be selectable via the local interface only with provisions for simple selection between local and remote interface. Should the special communications modes be required they may be easily selectable.

Another approach would permit the Cockpit Controller to modify COMM-1 and COMM-2 data and prevent the Cockpit Controller from modifying ECM/ESM data with the remote interface active. The second approach would require the ASQ-191 to accept inputs across both local and remote interfaces simultaneously. Conflicting inputs from both interfaces could

result.   The overhead involved in keeping track of valid inputs may prohibit the second approach.

Regardless of how the special communications modes are implemented, complete control of the ASQ-191 should not be possible from the aft cockpit and to create less impact on ALQ-99 software.

f.   ECM/ESM Scan Modes

Some changes undoubtedly will be required to the ECM/ESM scanning modes.   These changes will not affect the algorithms, but will affect data sent to the ALQ-99.   For example, receiver tuning data would assist the system operators in monitoring ASQ-191 activity on ALQ-99 displays. Receiver tuning data should be output in a format to be rapidly assimilated by the ALQ-99 for generation of "tuning carrots" on the Digital Display Indicator (DDI).   In addition, active signals intercepted by the ASQ-191 should be converted into Active Emitter Files for the ALQ-99.   [Ref. 7:p. 5] indicates the AYK-14 Tactical Computer should maintain and Active Emitter File for the ASQ-191.   Liaison with Mr. Pete Kantor at COMMATVAQWINGPAC NAS Whidbey [Ref. 9] indicates that Active Emitter Files from the ASQ-191 could be readily assimilated into the existing ALQ-99 software routines.

g.   Interrupts

The figures in [Ref. 8:pp. 16-17] represent the interrupt structure of the System Controller.   The most significant modification to the interrupt structure involves

the Mil-Standard 1553. The Remote Control Transmit Data and Remote Control Receive Data modules transmit and receive data across the remote interface. Both modules must be modified for the Mil-Standard 1553A to permit the system controller to function as a 1553 remote terminal.

Mil-Standard 1553 data words contain a 16 bit field into which the ASQ-191 and ALQ-99 will store higher level commands, status and data. Some of the commands from the ALQ-99 to the ASQ-191 must be assembled into proper format and error-checked before being passed on to the cognizant software module for action. The same type of protocol conversion is required to convert ASQ-191 data into 16 bit formats for transmission to the ALQ-99. The specifics for conversion must be determined through close liaison between Rockwell/Collins and PMTC Pt. Mugu.

h.   ESM Record Mode and Printer Commands

EA-6B operators may record critical ALQ-99 mission data for post-mission analysis. The ASQ-191 has similar capabilities, the ESM Mode and Printer commands, which, if modified may be able to support the EA-6B post-flight capability.

The ESM Mode of the ASQ-191 allows active target frequencies to be recorded directly into Target Tables in volatile memory. All active targets are recorded and the priority mode, if selected is disabled. In addition, specific types of signals may be recorded if a minimum and maximum time

for signal activity is specified. The ESM Mode requires some modification to prevent compromising TEAMS/ALQ-99 generated databases. For example, there are two different Target Tables for the Normal and Priority Search Modes. Target Tables will be loaded by the ALQ-99 operator across the 1553 interface. When the ESM Mode is selected Target Tables provided by the operator may be written over by the ASQ-191 and a mismatch in the ALQ-99 and ASQ-191 databases would occur. An option would be to modify the ASQ-191 ESM mode to write to the alternate nonvolatile Target Table (presumably Table #2), rather than the Target Table currently in use. If Active Targets were written into Table #2 then that data could be transmitted across the remote interface for recording on an RRS tape and later post-mission analysis.

The Printer Commands direct the ASQ-191 to print selected data via a hard copy printer located in the Cockpit Control. System Status, Target Tables and Active Targets may be output to the printer for later analysis. This same data may be useful for EA-6B post-mission analysis. The software modules which handle printer output would require modification to format data for, and output data to the Remote interface rather than the hardcopy printer.

3. Summary

The strategy has been to force the system at initialization to establish the 1553 interface with the ASQ-191 and load the initial operating parameters based upon

prefight mission planning data provided through TEAMS. Provisions are proposed to permit the operators to swap control between the front and rear cockpit as required and to permit graceful degradation of the 1553 interface. Normal ASQ-191 BIT should be selectable from and results returned to the ALQ-99. Status information must be exchanged between ASQ-191 and ALQ-99 and a subset of the existing status words may be sufficient to provide the ALQ-99 with required information. Protocol conversion must be performed to ensure coherent exchange of data across the 1553 bus. Provisions must also be made for post-mission analysis of ASQ-191 intercepts with TEAMS.

E. ALQ-99 SOFTWARE CONSIDERATIONS

The primary interface between the ALQ-99 system and the operator is the ALQ-99 Digital Display Group (DDG). The DDG group consists of two Digital Display Indicators (DDI), two associated Digital Display Indicator Controls (DDIC) and a Display Processor (DP). The DDI presents a large screen to the operator. The operator addresses specific area of the DDI by moving cursors to point to a specific location on the screen and then depressing keys on the DDIC. The DDI has a number of screen formats. One of the most frequently used format is the FREQ/AZ display which is separated into six distinct regions called zones. The software change proposals relating to the ALQ-99 primarily involve additions to the

63

information displayed to some of the unique displays and in some of the FREQ/AZ zones.

1. **Initialization Pages**

Prior to each flight the ALQ-99 operators initialize the ALQ-99 OBS system for a mission. OBS initialization data may be examined as required and changed during the mission. Three new initialization pages are proposed to incorporate the ASQ-191 the ALQ-99 System.

The first ASQ-191 System Initialization page permits operating parameters for the ASQ-191 to be examined and modified. The specific values for each field on the initialization page will be set at a TEAMS workstation and may be modified during initialization. Table 4 presents the fields required for the ASQ-191 Initialization page and ranges for each entry. The format for this page is flexible.

The second ASQ-191 initialization page will display in a table format the contents of the Normal Target Table #1 and the corresponding Limited Search Target Table currently maintained in nonvolatile memory in the ASQ-191. The third initialization page will display the contents of Priority Target Table #1 and its corresponding Limited Search Target Table. Both Tables will cross reference the associated ALQ-99 list in the following format:

1. List Number.

2. Symbol.

3. Frequency.

TABLE 4

ASQ-191 INITIALIZATION DATA

FIELD	VALUES
SCANNING FORMAT	(Normal, Selective, Priority, Blind)
LIMITED SEARCH	(Enabled, Disabled)
DATA LINK MODE	(Enabled, Disabled)
DATA LINK DWELL TIME	(50-4000)
GUARD MONITOR	(On, Off)
ALERT MONITOR	(On, Off)
ALERT FREQUENCY	(000.00 to 999.99)
SEARCH RANGE	(000.00 to 999.99)
TARGET TIME OUT	
ESM RECORD PARAMETERS	(ESM Targets, Active Targets, Total Target, Current Target, Priority Target, Limited Search)
ESM ACCEPTED ACTIVITY	(All, Hopping, Medium Hopping, Slow Hopping, Slow Hopping or Fixed Frequency, Fixed Frequency)
JAMMER TURN ON DELAY	(0 to 9999)
TARGET DROP-OFF DELAY	(0 to 5000)
CHANNELIZED DETECTION	(Enabled, Disabled)
SCAN STEP SEARCH SIZE	(25 to 1000)
FREQUENCY CHANGE DELAY	(150 to 400)
FM MODULATION	(30 to 9000)
REMOTE/LOCAL INTERFACE	(Remote or Local)
ALARM WORD PERSISTENCE	(0 to 9999)
SYSTEM RESET	(SELECT TO INITIATE)
ERASE NONVOLATILE MEM	(SELECT TO INITIATE)

4. Modulation.

5. Priority (Priority Table only).

Thus, the status of the ASQ-191 and associated libraries may be cross-referenced at a glance.

   2. **Bit Displays**

   The Normal ASQ-191 BITs are selected from the Cockpit Controller via the COMM-1 and COMM-2 pages. As proposed in the previous section the normal ASQ-191 BITs should be initiated from the aft cockpit as an option for the ALQ-99 OBS BIT. If the ASQ-191 BIT is selected both the COMM-1 and COMM-2 bits should be commanded across the 1553 remote interface. The results of the BITS should be displayed as simply as possible. If all of the COMM-1 and COMM-2 tests passed then "COMM-1 Passed" and "COMM-2 Passed" should be displayed as alerts in Zone Six of the DDI. If an item failed or was not tested, then an alert in Zone Six should indicate the specific item which failed. There are eight potential alerts which may appear as indicated in [Ref. 6:p. 32].

   3. **Freq/AZ Zone Two**

   Changes to Zone Two involve enhancing the frequency range to account for the increased coverage of the ASQ-191. The frequency scale in the extended range should be presented in megahertz. Manipulation of the displayed range should be identical to existing Zone Two operations.

   In order to indicate the current frequency an ALQ-99 receiver is scanning through "Tuning Carrots" are generated at

the corresponding frequency in Zone Two. The ASQ-191 should provide tuning data to the ALQ-99 in a format that may be easily assimilated by the AYK-14/DP and subsequently displayed on the DDI. The tuning rate of the ASQ-191 is so rapid that Tuning Carrots displayed at the DDI could lag behind the actual generate frequency of the ASQ-191. Rather than continuously generate Tuning Carrots, propose that a carrot be generated each time the receiver dwells on an intercept.

4. **Freq/AZ Zone Three**

ASQ-191 signal intercepts may be displayed in Zone Three similar to existing ALQ-99 Low Band intercept presentations. Symbology associated with the ASQ-191 intercept will be derived from the corresponding TEAMS generated library. Position in Zone Three will be based on intercept frequency. Intercept position in azimuth will be based upon some convention to be determined. One possibility is to display ASQ-191 intercepts centered in azimuth on the center of the DF Sector.

The ASQ-191 system has incorporated provisions to adjust the intensity and time period an intercept will remain on the ASQ-191 Cockpit Display. Due to the rapid scan rate of the ASQ-191 the same type of provision is required for display of ASQ-191 intercepts on the DDI. This feature should be loaded at system initialization and be selectable during the mission.

67

Manipulation of detected emitters in Zone Three should be as similar as possible to existing procedures. Protocol conversion will be required to convert DDG cursor position and keyboard entries into ASQ-191 commands. For example, receiver slews performed at the DDI, in ASQ-191 frequency ranges must be translated into an ASQ-191 "hold" keypad command and transmitted across the 1553 interface to the ASQ-191. Resume Scan must be similarly translated into an ASQ-191 "run" keypad command. Timing restrictions may require some modification of the ASQ-191 "Hold" algorithm.

Conventions for displaying active target frequencies, nontarget frequencies and other frequencies within the selected search range must be clear and unambiguous. For example, if the ASQ-191 is in the normal search mode and the Limited Search Mode option is selected two types of intercepts may appear. Active Targets should be displayed with normal symbology. Active nontarget frequencies should be displayed differently. An option would be to display the active nontarget symbol inside another symbol or display the nontarget with reduced intensity. (No provision exists to display active frequencies that are neither active target or non-target frequencies, although changes could be incorporated into the ECM/ESM scanning algorithms.)

When the Jam/Monitor (J/M) input is toggled to jam, jammer boxes should be displayed around the affected intercepts displayed in Zone Three. Boxes will be inhibited

from lock-out (non-jammed) frequencies. Thus Jammer Boxes
will provide a clear and unambiguous picture of jammed and
unjammed frequencies.

5. **Freq/AZ Zone Four**

Zone Four is important in that it is presented when
either the FREQ/AZ display or the GEO (Geographic) display is
generated. Zone Four should provide information that the
operator needs to know regarding the ASQ-191 status.
Recommend that "ASQ-191" appear in Zone 4 with an appropriate
alert or series of alphanumeric codes below it (X/X/XX/XX).
For example if the Remote/Local Switch was in local, the alert
"LOCAL" should be displayed to indicate that the aft cockpit
operators have no control of the system. If the Remote
interface is active, and initializing the "INIT" should be
displayed. After the ASQ-191 and ALQ-99 have successfully
established communications the current status of the system
should be displayed. A sample ASQ-191 status would appear as
"J/N/LS/REC." The sample could be translated according to the
following:

1. Status of J/M Switch i.e., "J" or "M."

2. Current Scanning Format...Normal = "N," Selective = "S,"
   Priority = "P" or Blind = "B."

3. Additional Scanning Options...Limited Search Mode =
   "LS," Data Link Mode = "DL," or neither mode = "XX."

4. ESM Record Mode...Mode Active = "REC," Mode Inactive =
   "XXX."

69

With the assumption that the most frequently selected and critical options are items 1 through 4 above, these options should be selectable by centering the cursors over the appropriate feature and depressing the Assign/Enter pushbutton on the DDIC. Each time Assign/Enter is pressed the selected feature should toggle to the next option.

When the J/M input is toggled to jam emitters, power out of the ASQ-191 transmitter should be displayed below the ASQ-191 status line.

6. <u>Zone Five</u>

Zone Five is used to display a wide range of information to the operator. The most important type of information is parametric data regarding intercepts and TEAMS generated libraries. Libraries, generated via TEAMS, command the ALQ-99 receivers to search specific frequency ranges. The ALQ-99 receivers search the frequency range until a signal is intercepted. The parametrics associated with the signal are passed through the ALQ-99 system, matched to an active library and displayed as Alarm Words in Zone Five.

In similar fashion ASQ-191 Libraries will directly correspond to search tables which command the ASQ-191 to search for specific frequencies. ASQ-191 Libraries will be loaded with ALQ-99 Libraries during a normal mission load. Simultaneously ASQ-191 Search Tables corresponding to the Mission Libraries will be loaded into the ASQ-191. Manipulation of the ASQ-191 Libraries should be similar to

manipulating ALQ-99 Libraries. ASQ-191 libraries should be activated/deactivated/modified in the same manner as existing ALQ-99 libraries. For example, deactivating ASQ-191 associated Libraries in Zone Five should delete the frequency contained in the library from the Target Table identified in the Library. The ALQ-99 should transmit the table deletion across the 1553 data bus to the ASQ-191 as a command to invoke the Target Table Editor and delete the frequency.

The parametric information associated with an ASQ-191 intercept should be identical to existing ALQ-99 generated Alarm Words with the exception of bearing. In order to distinguish between ALQ-99 and ASQ-191 intercepts recommend "COM" be substituted for bearing information. In order to direct the ALQ-99 to display ASQ-191 Alarm Words in Zone Five propose the operator place the cursors in the ASQ-191 frequency Range in Zone Two and depress the Assign Enter pushbutton. Expanded Zone Five, providing an expanded list of target tables or active communications frequencies/intercepts in Zone Three should be incorporated.

7. Zone Six

Zone Six is used to provide alerts to the operator regarding system status and contains an "Edit" line to display information entered on the keyboard. The integration of the ASQ-191 will require addition of a number of operator alerts. The following provides a summary of operator alerts which

could be incorporated into the system and the associated condition:

1. ASQ-191 COMM-1 or COMM-2 BIT--Results would be displayed as previously discussed in conjunction with the ALQ-99 OBS BIT.

2. Background BIT failures--Indicates that the System Controller has detected a fault in Bit 1 through 7 of the Receiver Status Word.

3. ASQ-191/ALQ-99 1553 Interface Failure--The EA-6B Tactical Computer is unable to establish/maintain communications with the ASQ-191 across the 1553 Interface.

4. Data Link--Indicates the ASQ-191 has detected a Data Link Frequency while operating in the Data Link Mode.

5. Guard--The ASQ-191 has detected a Guard Frequency Transmission.

6. Alert--The ASQ-191 has detected activity on a user selected "Alert Frequency."

7. Search Range--The search range of the current ASQ-191 Target Table is being modified.

8. ESM Record Mode--Indicates when the ESM Record Mode has been selected.

The ASQ-191 Operators Manual [Ref. 5] describes a number messages which result from operator input errors. Because error-checking routines should be incorporated into ALQ-99 and TEAMS software only a few of those errors may be required to include Loaded, Stored, Wait, No TGTS and No SRCH. The function of these messages is described in [Ref. 5:p. 22].

F. TEAMS SOFTWARE CONSIDERATIONS

TEAMS is utilized to conduct area planning, mission planning and postflight data reduction for EA-6B missions.

The data generated through TEAMS for mission planning is stored on an RRS tape and loaded into the ALQ-99 system prior to flight. Inflight ALQ-99 information may be stored on the same tape for post-mission analysis. To support ASQ-191 integration, TEAMS will conduct error checking of ASQ-191 initialization data, generation of ASQ-191 Libraries and development of Target Tables from Library Data. Changes will be required to TEAMS software to permit the inclusion of Communication Location, Emitter Characteristics Data and Communications Jammer Technique Data into the TEAMS Database. Proprietary information indicates that this may not be difficult. However, questions regarding the compartmentalization of Communications intelligence and other security considerations must be addressed and resolved before complete integration is possible.

TEAMS provides an excellent means to initialize the ASQ-191. The ASQ-191 Initialization data and pages proposed in the previous section should also be available at TEAMS. Operators should be permitted to edit operating parameters and target tables in their entirety. TEAMS will provide an excellent capability to carefully plan and standardize ASQ-191 operating parameters.

Libraries generated via TEAMS will correspond to communications threats located in the mission area of interest. TEAMS will generate a series of libraries and Normal, Priority and Selective Target Tables. ALQ-99 and

73

ASQ-191 libraries will be stored on an RRS mission tape along with the appropriate Target Tables. Here TEAMS will provide a significant improvement to the existing ASQ-191 capabilities. The ASQ-191 will only permit two Normal and Priority Target Tables to be stored in nonvolatile memory at a time. RRS Tapes developed during mission planning may have as many target tables as the RRS Tape can hold. In addition these tables may be rapidly reprogrammed inflight to support real time missions.

ASQ-191 Libraries will need to include the following information, at a minimum:

1. Discrete Search Frequency.

2. Alphanumeric Symbol (Displayed in Zone Three).

3. Associated Target Table (Normal or Priority or both).

4. Priority (if associated with Priority Table).

5. Limited Search Status.

6. Optimum and possible baseline jamming techniques.

Careful management of ASQ-191 Libraries is required because of the extremely large number of discrete frequencies for which the ASQ-191 may programmed to search for. Each Target Table is capable of storing all frequencies over the entire frequency coverage of the ASQ-191. The requirement to have thousands of ASQ-191 libraries does not exist, but the total number of ALQ-99 libraries may need to be expanded. With the capability to rapidly load and reload target tables

an ALQ-99 restriction on the upper limit of libraries should not severely impact ASQ-191 employment.

The ASQ-191 will be required to provide postflight information across the 1553 interface for storage on an RRS tape. The TEAMS will be required to read and interpret this data for post-mission analysis.

VII. CONCLUSIONS

The first portion of the thesis was focused upon a practical application of formal modeling techniques to a commonly used Mil-Standard. **Systems of Communicating Machines** is a formal model useful for describing communications protocols. SYSCOM may model any protocol with three parts which include Finite State Machine diagrams, shared and local variables and predicate action tables. After the protocol has been modeled, the correctness of the model is verified through reachability analysis. The analysis should show that the model is error free, within restrictions placed upon the system by the associated hardware.

SYSCOM was used to specify the Mil-Standard 1553. Assumptions were made to simplify the model and promote understanding. The bus controller and remote terminals were described with their own unique finite state machines, local variables and predicate action table. The data bus itself was modeled with shared variables. Although the model alone is sufficient to fully describe the protocol, additional explanation was provided to describe how features of the Mil-Standard were incorporated.

Reachability Analysis was conducted to prove correctness of the model. Analysis began from a global starting state and proceeded to exhaustively analyze all possible states. Three

deadlocks were identified. The deadlocks were not significant because they were the result of limitations of the model to fully describe the 1553 hardware.

After a simplified model was presented and analyzed a more complete model was developed. Timing requirements, error conditions and the 1553 broadcast mode were added to the model. A methodology for building a more complex model was presented.

Formal Specification is useful for understanding protocols, proving correctness and implementation in software and hardware. The SYSCOM model may be utilized to demonstrate the utility of formal specification.

This thesis has used the SYSCOM model to represent the widely utilized Mil-Standard 1553. The finite state machine diagram presents a clear and precise abstraction of the model. The predicate action and finite state machine in conjunction have been used to demonstrate the workability and correctness of the protocol by reachability analysis. Finally the predicate action table may be used to convert the model directly into software or hardware. **Systems of Communicating Machines** has many other practical applications to both civilian and military applications. Token Ring and Ethernet protocols have already been specified. The Mil-Standard 1760A which describes Mission Stores Interface between aircraft and weapons could also be modeled with SYSCOM as could other protocols in development.

The final chapter of the thesis was focused on the requirements to integrate the ASQ-191 Radio Countermeasures Set into the EA-6B ICAP II P-99 and Block 86 Prowler Aircraft. A description of the ASQ-191 was provided. A proposal for how the ASQ-191 and EA-6B systems would integrate was proposed. Specific changes to the ASQ-191, ALQ-99 and TEAMS systems were also proposed.

The benefits resulting from the proposed integration are numerous. The most important being the improvement in capability of the EA-6B community to employ the ASQ-191 as a fully integrated communications countermeasures capability for intercept/exploitation and jamming applications. Mission planning will include the ASQ-191 as an integral part of the aircraft weapons system and ASQ-191 initialization will be rapidly accomplished with reduced potential for operator error. Inflight utilization of the ASQ-191 will not be limited to two target tables but to the number of missions an RRS tape can accommodate. Complete utilization of ASQ-191 ECM/ESM capabilities will be possible from the aft cockpit and safety of flight will not be compromised in the front cockpit. Post-mission data reduction will permit enhancement of the TEAMS database and improve ASQ-191/EA-6B employment in subsequent missions.

Further research is required to finalize detailed requirements for transparent integration of the ASQ-191 into the EA-6B Weapons System. The proposed software change

requirements for the ASQ-191 are not all exhaustive but present a basis for further development. EA-6B C3CM requires full integration of the ASQ-191 for effective passive ESM/COMINT and smart communications jamming. Close cooperation will be required between the engineers at Rockwell/Collins, PMTC Pt. Mugu and PRB Corporation to successfully integrate the systems. Most importantly the support of the EA-6B community and NAVAIRSYSCOM is required, without which this integration will not be completed.

LIST OF REFERENCES

1. Department of Defense, Military Standard 1553B, **Aircraft Internal Time Division Command/Response Multiplex Data Bus**, 12 February 1980.

2. Pinkowitz, D.C., "Mil-Std-1553B: The Military Standard for Avionics Integration," **EW Design Engineers' Handbook**, Horizon House Microwave, 1988.

3. Lundy, G.M., and Miller, R.E., "A Variable Window Protocol Specification and Analysis," **Proceedings of the Eighth International Symposium on Protocol Specification, Testing and Verification** (IFIP WG 6.1), Atlantic City, New Jersey, June 7-10, 1988.

4. ILC Data Device Corporation, "Mil-Std-1553 Designer's Guide," Second Edition, ILC Data Device Corporation, 1988.

5. Rockwell International, **Operators Manual AN/ASQ-191A Radio-Countermeasures Set**, Collins Defense Communications Rockwell International Corporation, Cedar Rapids, Iowa, March 1987.

6. Rockwell International Corporation Collins Defense Communications, **Software Requirements Specification AN/ASQ-XXX Radio Countermeasures Set System Controller CSE**, Rockwell International, 1989.

7. Pacific Missile Test Center EA-6B Systems Division, **1553A CRE and ASQ-191 Integration**, by Glen Wheeler, Ralph Geirut et al., July 1989.

8. Rockwell International Corporation Collins Defense Communications, **Software Design Document AN/ASQ-XXX Radio Countermeasures Set System Controller CSE**, Rockwell International, 1989.

9. Interview between Mr. P. Kantor, COMMATVAQWINGPAC NAS Whidbey and the author, November 7, 1989.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 0142   2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Director for Command, Control, and
   Communications Systems, Joint Staff   1
   Washington, D.C. 20318-6000

4. Superintendent, Code 52   1
   Computer Science Department
   Naval Postgraduate School
   Monterey, California 93943-5000

5. Superintendent, Code 012   1
   Naval Postgraduate School
   Monterey, California 93943-5000

6. Professor Gilbert M. Lundy, Code 52Ln   3
   Computer Science Department
   Naval Postgraduate School
   Monterey, California 93943-5000

7. CDR James R. Powell, Code 62Pl   3
   Electronics and Computer Engineering
   Department
   Naval Postgraduate School
   Monterey, California 93943-5000

8. Captain Dana McKinney   1
   Attn: PMA-234-1
   Naval Air Systems Command
   Washington, D.C. 20361

9. Commander, Naval Air Systems Command   1
   Attn: LCDR L. Draper, PMA-234-2
   Washington, D.C. 20361

10. Commander, Naval Air Systems Command   1
    Attn: CDR John D. Langford
    Washington, D.C. 20361

11. Commanding Officer   1
    Attn: EA-6B Project Officer
    Air Test and Evaluation Squadron Five
    Naval Weapons Center
    China Lake, California 93555

12. LCDR Steve Ewell   3
    Attn: Code 33
    Medium Attack Tactical Electronic
       Warfare Wing, Pacific
    NAS Whidbey Island
    Oak Harbor, Washington 98278

13. Commanding Officer   2
    Tactical Electronic Warfare Squadron 129
    NAS Whidbey Island
    Oak Harbor, Washington 98278

14. Commanding Officer   1
    Tactical Electronic Warfare Squadron 130
    NAS Whidbey Island
    Oak Harbor, Washington 98278

15. Commanding Officer   1
    Tactical Electronic Warfare Squadron 131
    AS Whidbey Island
    Oak Harbor, Washington 98278

16. Commanding Officer   1
    Tactical Electronic Warfare Squadron 132
    AS Whidbey Island
    Oak Harbor, Washington 98278

17. Commanding Officer   1
    Tactical Electronic Warfare Squadron 133
    AS Whidbey Island
    Oak Harbor, Washington 98278

18. Commanding Officer   1
    Tactical Electronic Warfare Squadron 134
    AS Whidbey Island
    Oak Harbor, Washington 98278

19. Commanding Officer   1
    Tactical Electronic Warfare Squadron 135
    AS Whidbey Island
    Oak Harbor, Washington   98278

20. Commanding Officer   2
    Tactical Electronic Warfare Squadron 136
    FPO San Francisco   96601

21. Commanding Officer   1
    Tactical Electronic Warfare Squadron 137
    AS Whidbey Island
    Oak Harbor, Washington   98278

22. Commanding Officer   1
    Tactical Electronic Warfare Squadron 138
    AS Whidbey Island
    Oak Harbor, Washington   98278

23. Commanding Officer   1
    Tactical Electronic Warfare Squadron 139
    AS Whidbey Island
    Oak Harbor, Washington   98278

24. Commanding Officer   1
    Tactical Electronic Warfare Squadron 140
    AS Whidbey Island
    Oak Harbor, Washington   98278

25. Commanding Officer   1
    Tactical Electronic Warfare Squadron 141
    AS Whidbey Island
    Oak Harbor, Washington   98278

26. Commanding Officer   1
    Tactical Electronic Warfare Squadron 142
    AS Whidbey Island
    Oak Harbor, Washington   98278

27. Commanding Officer   2
    VMAQ-2
    MCAS Cherry Point, North Carolina   28533

28. Commander, Pacific Missile Test Center   1
    Attn: Mr. Glen Wheeler, Code 4041
    Point Mugu, California   93042

29. Commander, Pacific Missile Test Center   1
    Attn: LCDR Don Marcotte
    Point Mugu, California 93042

30. Commander, Pacific Missile Test Center   1
    Attn: Mr. Ralph Gierut
    Point Mugu, California 930-42

31. Applied Physics Laboratory   1
    Attn: Mr. Dan Henderson
    Johns Hopkins University
    1110 Johns Hopkins Road
    Laurel, Maryland 20810

32. Commander, Joint Electronic Warfare Center   1
    Attn: LCDR J. Coulson, Deputy Commander
    San Antonio, Texas 78243

33. Commanding Officer   1
    Attn: LCDR Peter H. Christensen
    Tactical Electronic Warfare Squadron 129
    NAS Whidbey Island
    Oak Harbor, Washington 98278

34. Collins Defense Communications   3
    Attn: Mr. Jack J. Hill
    Rockwell International Corporation
    350 Collins Road NE.
    Cedar Rapids, Iowa 52498